Savitribai Phule Pune University

(Formerly University of Pune)

Two Year Post Graduate Degree Programme
M. Sc. (Industrial Mathematics with Computer Applications)

(Faculty of Science & Technology)

New Syllabi

For

M. Sc. (IMCA)

(For Colleges Affiliated to Savitribai Phule Pune University)

(As per National Education Policy- 2020)

To be implemented from the Academic Year 2023-2024

❖ Preamble

The board of studies in Mathematics of Savitribai Phule Pune University, Pune made a rigorous attempt to revise the curriculum of postgraduate programmes M.Sc.(Industrial Mathematics with Computer Applications) to align it with National Education Policy-2020 and UGC Quality Mandate for Higher Education Institutions-2021. The process of revamping the curriculum started with the series of Meetings, workshops, webinars and discussions with sub-committees conducted by the University to orient the teachers about the key features of the Education Policy, enabling them to revise the curriculum in sync with the Policy. Appropriate orientation of the faculty about the vision and provisions of NEP-2020 made it easier for them to appreciate and incorporate the vital aspects of the Policy in the revised curriculum focused on 'creating holistic, thoughtful, creative and well-rounded individuals equipped with the skill sets of 21st century for the 'development of an enlightened, socially conscious, knowledgeable, and skilled citizen of the nation'.

With NEP-2020 in background, the revised curricula will articulate the spirit of the policy by emphasizing upon- integrated approach to learning; innovative pedagogies and assessment strategies; multidisciplinary and Interdisciplinary education; creative and critical thinking; student-centric participatory learning; imaginative abilities and flexible curricular structures to enable creative combination of disciplines for the study. The Credit framework for designing Post Graduate Programmes prepared by the University as per the guidelines of State Government is followed as it is and the curriculum is further modified as per the needs specified in NEP. The curriculum is developed to trigger the inquisitiveness, discussion, analytical ability and quest for discovery among learners. Mathematics is a powerful tool for understanding and communicate globally that organizes our lives and prevents chaos, which helps us understand the world and provides an effective way of building mental discipline. Along with mathematical skills, it is also expected that students will learn life skills like argumentation, communication and general social values which are necessary to life rich, productive and meaningful life. Additionally, the knowledge of mathematical modelling and computational training which the students acquire during the Programme makes them highly sought after. In keeping with the demands of industry and academia, the syllabus is updated regularly, with inputs taken from various stakeholders including students, alumni and parents at different stages of the modification/addition of the syllabus. The new curriculum provides a synoptic overview of possible career paths mapped by a postgraduate degree in Mathematics Teaching, Research, Engineering, Computer programming, Statistician, Competitive examination, and many more.

Important Highlights

(1) **Title of the Programme** : M. Sc. (Industrial Mathematics with Computer Applications)

(2) **Duration**: 02 years (Four semesters) Full-time Post - Graduate Degree Programme

(3) **Intake Capacity**: 30 students

(4) **Total number of credits**: 88 credits

(5) **Programme Structure of M. Sc. (IMCA)**: For M. Sc. (IMCA) Degree, a student has to earn the minimum 88 credits from at least FOUR semesters. The structure of the programme is as follows:

(a) In each of the four semesters I, II, III, and IV, the Department will offer at least 22 credits.

(b) In each semester, there will be three mandatory courses each of 4 credits, and one elective course. Also, in each of the semesters I, II, and III, there will be a mandatory course of 2 credits.

(c) Each course of 4 credits, other than OJT and RP.

(d) A student has to attend 1-hour classroom teaching per week for one credit of theory and 2 hours lab work/problem-solving session/ related activities per week for one credit of practical.

(e) Practical sessions (lab work/problem-solving session/related activity) will be conducted in batches. A batch for such sessions will be of size maximum of 12 students.

(f) The Department may conduct necessary lectures/workshops as a part of OJT.

(I) Each course of 4 credits (T + P) will carry 100 marks and the evaluation of the course will be carried out by considering T and P Separately. There will be Continuous Assessment (CA) and End Term Examination for each course.

(II) The CA will be based on minimum two internal tests (IT). In addition, a teacher may consider one or more of the following.

(i) Home Assignment(s)

(ii) Seminar/Presentation (Individual / Group)

(iii) Laboratory assignment

(iv) Group Discussions / Oral

(v) Research Paper Review

(vi) Technology Demonstration

(III) For passing a course, a student has to score a minimum of 40% marks in each of the CA and ETE separately and a minimum of 40% marks in the combined grading of CA and ETE. If a student fails to score a minimum of 40% marks in CA in a course, then the result of such a course will be FAIL.

(IV) For both OJT and RP, the CA will be based on grades awarded by guide/mentor while the ETE will be based on presentation/oral/discussion/ any other criterion decided by

Sem I-Research Methodology (RM) - 4 credits
Sem II-On Job Training (OJT) - 4 credits
Sem III - Research Project (RP1) - 4 credits
Sem IV- Research Project (RP2)- 6 credits.

(6) **Exit Option:** After successful earning of 44 credits offered by the Department for the first two semesters (First year-I, II Sem), a student will have the option of exit from the programme. In this case, the student will be conferred with PG Diploma in Mathematics.

(7) **ATKT Rules**: A student who wishes to take admission to the second year (register for third or fourth semester) of M. Sc. (IMCA) programme must have earned at least 22 credits from the total credits of two semesters of the first year of M. Sc. (IMCA).

(8) **Research Project (RP-1 & RP-2):**

Procedures and guidelines for the conduct of the Research project:

(a) A student is supposed to register for the course RP-1 and RP-2 separately in a group of 2 to 4 students.

(b) A student will carry out the academic activity for the course throughout the semester.

(c) The course is to be completed under the supervision and guidance of a teacher. Each teacher of the Department of Mathematics, Savitribai Phule Pune University is expected to guide at least one group of students.

(d) The respective teacher is expected to engage a group of students for at least 4 hours/week for RP-1 and at least 6 hours/week for RP-2.

(e) Every group will submit a dissertation at the end of the semester duly signed by all group members and the respective teacher.

(9) **On Job Training (OJT)** In this course, the students are expected to do the On Job Training (OJT) in appropriate Industries/Government sectors/Institute etc. to get hands on experience. The department may conduct necessary lectures/workshops/seminars as a part of OJT. The course will be conducted as per the guidelines of the College/ University and Government of Maharashtra.

10) **Eligibility:** B. Sc./ B. E./ B. Tech.(Mathematics subject at least at second year).

Programme Outcomes (POs)

Name of the Programme: M.Sc.(IMCA)

PO-No.	Programme Outcomes *The Student will be*	Component
PO-1	Capable of delivering basic disciplinary knowledge gained during the programme.	Basic Knowledge
PO-2	Capable of describing advanced knowledge gained during the programme	In-depth Knowledge
PO-3	Able to gain knowledge with the holistic and multidisciplinary approach across the fields.	Holistic and multidisciplinary Education
PO-4	Capable of analyzing the results critically and applying acquired knowledge to solve the problems	Critical thinking and Problem-Solving abilities
PO-5	Capable to identify, formulate, investigate and analyze the scientific problems and innovatively design and create product solutions to professional and real life problems.	Creativity and innovation
PO-6	Able to develop a research aptitude and apply knowledge to find the solution of burning research problems in the concerned and associated fields at global level.	Research aptitude and global Competency
PO-7	Able to Learn interdisciplinary and multidisciplinary skill sets and advanced techniques to apply them for better livelihood of mankind.	Skills enhancement
PO-8	Able to learn and work in a groups and capable of leading a team even.	Leadership and Teamwork abilities
PO-9	Able to acquire lifelong learning skills which will lead important to better opportunities and improve quality of life.	Environmental and human health awareness
PO-10	Inculcate the professional and ethical attitude and ability to relate with social problems.	Ethical thinking and Social awareness
PO-11	Capable to establish independent start-up/innovation Centre etc.	Lifelong learning skills and Entrepreneurship

Programme Specific Outcomes (PSOs)

Sr. No.	**Programme Specific Outcomes** *The student-*
PSO-1	will have a strong foundation in both pure and applied mathematics.
PSO-2	will have the knowledge of the fundamental axioms in mathematics and capability of developing ideas based on them and inculcate mathematical reasoning.
PSO-3	will be able to apply mathematical skills for solving problems and can prepare himself for various competitive exams.
PSO-4	will acquire the knowledge of a wide range of mathematical techniques and application of mathematical methods/tools in science, social science, engineering and technology
PSO-5	will have basic knowledge of programming and computational techniques as required for employment.
PSO-6	will be able to develop analytical skills, critical thinking, creativity, communication, and presentation skills through assignments, seminars, Training, and Research project.

Course Structure of the Programme: M. Sc.(IMCA) Part- I
Approved by B.O.S.

Year	Level	Sem.	Course Type	Course Code	Course Title	Remark	Credit	No. of Hours
1	6.0	I	Core	IMT-411	Linear Algebra	Theory & Practical	**4** (2T+2P)	30+60
			Core	IMT-412	Discrete Mathematical Structure	Theory	**2**	30
			Core	IMT-413	Object Oriented Programming using C++	Theory	**2**	30
			Core	IMT-414	Data Base Management System	Theory	**2**	30
				IMT-415	Lab on C++ and DBMS	Practical	**4**	120
			Elective	IMT-416(A)	Statistical Methods	Theory & Practical (Choose any one)	**4** (2T+2P)	30+60
				IMT-416(B)	Numerical Analysis			
				IMT-416(C)	Operations Research			
			RM	IMT-417	Research Methodology	Theory & Practical	**4** (2T+2P)	30+60
		II	Core	IMT-421	Foundation of Analysis	Theory	**4**	60
			Core	IMT-422	Applied Algebra	Theory	**4**	60
			Core	IMT-423	Data Structure	Theory	**2**	30
			Core	IMT-424	Java Programming	Practical	4	120
			Elective	IMT-425(A)	Web Technology	Theory /Practical (Choose any one)	**4** (2T+2P)	30+ 60
				IMT-425(B)	Financial Mathematics			
				IMT-425(C)	Computational Geometry			
			OJT/FP	IMT-426	On Job Training or Field Project	Practical	**4**	120

Course Structure of the Programme: M. Sc.(IMCA) Part- II

Year	Level	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of Hours
2	6.5	III	Core	IMT-531	Ordinary Differential Equations	Theory	**4**	60
			Core	IMT-532	Computer Networks	Theory	**2**	30
			Core	IMT-533	Design and Analysis of Algorithm	Theory	**4**	60
			Core	IMT-534	Data Analysis Using Python	Practical	**4**	120
			Elective	IMT-535(A)	Data Mining	Theory (Chooseany one)	**4** (2T+2P)	30+60
				IMT-535 (B)	Dot Net			
				IMT-535 (C)	Statistical Inference			
			RP	IMT-536	Research Project	Practical	**4**	120
		IV	Core	IMT-541	Software Engineering and Testing	Theory	**4**	60
			Core	IMT-542	Optimization Techniques	Theory	**4**	60
			Core	IMT-543	Artificial Intelligence and Machine Learning	Practical	**4**	120
			Elective	IMT-544(A)	Go Programming	Theory (Chooseany one)	**4** (2T+2P)	30+60
				IMT-544(B)	DEFI Block chain Technology			
				IMT-544(C)	Graph Theory			
			RP	IMT-545	Research Project	Practical	**6**	180

Details of Syllabus:

Semester-I

IMT- 411: Linear Algebra [2(T) +2(P)=04 Credits]

Course Objectives:

- To develop a solid understanding of vector spaces, linear transformations, and their properties.
- To comprehend the concept of orthogonality and its applications, including projection methods and the Gram-Schmidt process.
- To study positive definite matrices, their properties, tests for positive definiteness, singular value decomposition, and quadratic forms.
- To analyze various applications of linear algebra in image processing, computer graphics, pattern recognition, and the Google PageRank algorithm.

Course Outcomes:

- Determine key concepts associated with Vector Spaces Illustrate the various properties of Determinant Function and solve examples justifying the properties
- Apply concept of Orthogonality to find an Orthogonal Bases using Gram Schmidt Identify the role of Eigenvalues and Eigenvectors in Matrix Decompositions.
- Determine the concept of Positive Definite Matrices and build sophisticated principles to apply it in various applications
- Determine the concept of Positive Definite Matrices and build sophisticated principles to apply it in various applications
- Describe key applications of the Course which have useful implications in Applied Sciences which include Linear Programming, Networks and Game Theory

Course Content

Unit 1. Vector Spaces **[15 Hours]**

1.1 Vector spaces and subspaces
1.2 Solving homogeneous and nonhomogeneous systems
1.3 Linear Independence
1.4 Basis and dimension
1.5 Four Fundamental Subspaces
1.6 Linear Transformations
1.7 Applications to Graphs and Networks

Unit 2. Orthogonality **[15 Hours]**

2.1 Orthogonal Vectors and Subspaces
2.2 Cosines and Projections on to lines
2.3 Projections and Least Squares

2.4 Orthogonal Bases and Gram-Schmidt

2.5 The Fast Fourier Transform

Unit 3. Determinants **[9 Hours]**

3.1 Introduction

3.2 Properties of Determinants

3.3 Formulas for the Determinant

3.4 Applications of the Determinant

Unit 4. Eigenvalues and Eigenvectors **[15 Hours]**

4.1 Introduction

4.2 Diagonalization of a matrix

4.3 Difference equations and powers of a matrix

4.5 Differential Equations and Matrix Exponentials

4.6 Complex Matrices

4.7 Similarity Transformations

Unit 5. Positive Definite Matrices **[15 Hours]**

5.1 Minima

5.2 Maxima and Saddle Points

5.3 Various Tests for Positive Definiteness

5.4 Singular Value Decomposition along with various applications

5.5 Minimum Principles

5.6 The Finite Element Method

Unit 6. Computation with Matrices **[12 Hours]**

6.1 Introduction

6.2 Matrix Norm and Conditional number

6.3 Computation of Eigenvalues

6.4 Iterative methods for solving linear systems Applications

Unit 7. Applications **[9 Hours]**

7.1 Google Page Ranking Algorithm

7.2 Principal Component Analysis

7.3 Pattern reorganization in Signal Processing

Recommended Book:

1. Linear Algebra and its Applications, Gilbert Strang, Fourth Edition (Chapter: 2 - 7)

Reference Books:

1. Elementary Linear Algebra (Applications Version), Howard Anton, Chris Rorres, Wiley Publications
2. Linear Algebra and its Applications, David Lay, Third Edition, Pearson Publications
3. Linear Algebra done Right, Sheldon Axler, Springer Publications
4. Linear Algebra, Kenneth Hoffman, Ray Kunze, MIT Press.

*****

IMT- 412: Discrete Mathematical Structure [02 Credits]

Course Objectives:

- Describe the propositional equivalences, quantifiers, predicates and different types of proofs, Articulate basic concepts of Logic
- Define basic concepts of Graph theory with focus on key concepts associated with graphs. Illustrate various mathematical properties of graphs and solve examples to justify the properties. Compare different types of graphs and operations on graphs.
- Apply basic counting principles and for solving problems based on arrangements and selections
- Understand the notion of generating functions and calculate the coefficients of generating functions. Formulate recurrence relations to solve problems. solve certain first order and second order recurrence relations.
- Conclude application areas of Discrete Mathematics.
- Develop an ability to solve individually and creatively advanced problems connected with its applications to Discrete Mathematics.

Course Outcomes:

- Understanding basic concepts of discrete Mathematical structures.
- Development of critical thinking for the Mathematics related to computer science.
- Application and construction of algorithms to solve problems in Discrete mathematics and in computer applications.
- Evaluate combinations and permutations on sets.
- Able to analyse logical propositions via truth tables.
- Able to model and solve real world problems using graphs and trees.

Course Content

Unit 1. Mathematical Logic [7 Hours]

1.1 Propositions (Statements), Logical connectivity's, Compound statements form, truth tables, tautology, implications and equivalence of statements forms, logical identities.

1.2 Normal forms: disjunctive normal form and, simplification. Conjunctive normal form, logical implications, valid arguments, methods of proof.

1.3 Theory of inference of statement calculus, predicate calculus, qualifiers free and bound variables, theory of inference of predicate calculus

Unit 2. Introduction to Graphs [8 Hours]

2.1 Basic terminology, simple and weighted graph, adjacency and incidence, hand-shaking lemma, underlying graph of a digraph

2.2 Complete graph, regular graph, bipartite graph, complete bipartite

2.3 Isomorphism, complement of graph, connected graphs, paths-simple, elementary, circuit – simple

2.4 Elementary Edge connectivity, vertex connectivity

2.5 Eulerian path and Eulerian circuit, planar graph Euler's formula.

Unit 3. General Counting Methods for arrangements and selections [8 Hours]

3.1 General Counting Methods for arrangements and selections
3.2 Addition Principle, Multiplication Principle
3.3 inclusion and Exclusion principle and problems based on these basic principles
3.4 Simple arrangements and selections
3.5 Arrangements and selections with repetitions.

Unit 4. Generating functions and Recurrence relations [7 Hours]

4.1 Generating function models
4.2 calculating coefficients of generating functions
4.3 Recurrence relation models
4.4 solving linear recurrence relations
4.5 Introduction to divide and conquer recurrence relation.

Reference Books:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications (TATA McGraw – HILL), Edition 6
2. John Clark and Derek Allan Holton, A first look at Graph Theory
3. Alan Tucker Applied Combinatorics, John Willey, Fourth Edition.

Additional References:

1. Kolman, Busby, Ross and Rehman, Discrete Mathematical Structures, Pearson Edition Sixth Edition
2. N. Deo, Graph theory with Applications to Computer Science and Engineering, PHI publication.
3. Douglas B. West, Introduction to Graph Theory, Pearson Education, Second Edition
4. Purna Chandra Biswal, Discrete Mathematics and Graph Theory, Fourth Edition (PHI.)
5. V. Krishnmurthy, Combinatorics Theory and Applications, East-West Press Pvt Ltd.

******

IMT- 413: Object Oriented Programming Using C++ [02 Credits]

Course Objectives:

- To understand the fundamental Concepts of programming.
- To solve simple and complex problems using C++.
- To understand the implementation of various data structures and algorithms.
- To understand the Object-Oriented Programming concepts using the C++ language.
- To understand the concept of I/O files & exception handling
- To solve the real-world scenarios using top-down approach.

Course Outcomes:

- To understand the principles of data abstraction, inheritance and polymorphism
- To understand the concept of function overloading, operator overloading, virtual functions and polymorphism.
- Apply the principles of virtual functions and polymorphism.
- Demonstrate the use of various OOPs concepts with the help of programmes
- Evaluate the I/O Introduces exception handling
- Creating programmes using classes and objects in C++.

Course Content

Unit 1. Introduction to C++ [2 Hours]

1.1 History of C & C++?
1.2 Features of C++.
1.3 Procedure-oriented programming
1.4 OOP vs. procedure-oriented programming
1.5 Starting with a simple "Hello World" programme Compiling, linking and running a C++ programme

Unit 2. Basics of C++ Programming [5 Hours]

2.1 Data types, Declaration of variables, Expressions
2.2 Operators, Operator Precedence, Evaluation of expressions,
2.3 Input/output function
2.4 Decision Control statement: - if ,if-else, switch-case
2.5 Loop Control Structure - While, for & do-while Jump statement – Break & Continue

Unit 3. Function [3 Hours]

3.1 User defined functions: - declaration , definition, function call, parameter passing (by value), return statement.
3.2 Concept of Recursive functions.
3.3 Scope of variables and Storage classes(local, global & static)

Unit 4. Array and Pointer **[4 Hours]**

4.1 Concept of One, Two and Multidimensional array.
4.2 Array Operations - declaration, initialization, accessing array elements.
4.3 Concepts of Pointer , initialization & declaration the pointer, pointer arithmetic, array of pointer.

Unit V.: C++ Classes & Objects **[6 Hours]**

5.1 Basics of Object-Oriented concepts (Encapsulation, Abstraction, Data Binding, Inheritance, Polymorphism)
5.2 Difference between structure & class
5.3 Classes and Objects
5.4 Classes and Access Specifiers
5.5 Defining data members and member functions
5.6 Array of objects Types of constructors & Destructors

Unit 6. Inheritance **[4 Hours]**

6.1 Introduction
6.2 Types of Inheritance
6.3 Base class and derived class examples
6.4 Virtual base class Abstract class

Unit 7. Function Overloading and Friend Function **[4 Hours]**

7.1 Concepts of Polymorphism - Static and Dynamic binding
7.2 Function overloading & overriding
7.3 Virtual functions and pure virtual functions
7.4 Concept of Friend Function
7.5 Operator Overloading

Unit 8. Files and Streams **[2 Hours]**

8.1 Classes for file stream operations(ifstream, ofstream, fstream)
8.2 Opening and closing a file
8.3 Input and Output function
8.4 File updation with random access

Reference Books:

1. Object Oriented Programming (C++) – Balaguruswamy

2. The Complete Reference C++ by Herbert Schildt

3. Object Oriented Programming with C++ by Robert Lafore

4. Object Oriented Programming with C++, Mahesh Bhave, Sunil Patekar Pearson Publication

Additional References:

Free Online course as a part of assignment can be given to student.

1. Online course from Great Learning Free Courses on C++
2. Udemy Online free Course on C++ for Beginners & Advance

****

IMT- 414: Database Management System [02 Credits]

Course Objectives:

- Discuss the fundamental Concepts of Databases.
- To understand Entities, attributes, relationships, constraints etc.
- To understand the implementation of various databases using SQL.
- To understand the Transaction and introduction to NOSQL Databases
- Familiarize the students with a good formal foundation on the relational model.
- To understand the fundamental concepts of PLSQL

Course Outcomes:

- Understand working of DBMS.
- Identify Entities, attributes, relationships, constraints and Draw the ER Diagram.
- Implement appropriate database for computer-based systems according to the user requirements, appropriate syntax to write SQL commands to perform various RDBMS operations
- Introduction to NOSQL Databases
- Understand Model Entity-Relationship diagrams for enterprise level databases
- Implement PLSQL on appropriate database.

Course Content

Unit 1. Introduction to DBMS and RDBMS [3 Hours]

1.1 Data, Information, Database,
1.2 DBMS, DBMS Architecture,
1.3 RDBMS, Advantages and Disadvantages of RDBMS,
1.4 Entities, Attributes, constraints, Keys.
1.5 ER Diagram and its graphical notations
1.6 DBA, Role of DBA

Unit 2. Relational Algebra [3 Hours]

2.1 Concept
2.2 Select, Project, Union, Intersection, Set Difference, Cartesian Product, Rename

Unit 3. SQL [12 Hours]

3.1 DDL, DML, DCL, TCL
3.2 Commands and their syntax in DDL and DML,Primary key, Foreign key, Unique key, constraints
3.3 Queries, Multi-table Retrievals, Nested Queries, Complex Queries, Aggregate Functions & views
3.4 Operators in SQL Joins

Unit 4. Relational Database Design Using PLSQL [7 Hours]

4.1 Introduction to PLSQL

4.2 PL/PgSqL: Datatypes, Language structure

4.3 Controlling the programme flow, conditional statements, loops

4.4 Stored Procedures & Functions

4.5 Cursors Triggers

Unit 5. Introduction to NoSQL Databases [5 Hours]

5.1 Concept,

5.2 Key features

5.3 Advantages and disadvantages

5.4 Types of NoSQL Databases

5.5 When NoSQL databases should be used

5.6 Difference between SQL and NoSQL

5.7 Applications using NoSQL Databases.

5.8 Introduction to MongoDB

References and WEB References:

1. Raghu Ramakrishnan, Johannes Gehrke: Database Management Systems, ISBN: 9780072465631, TMH
2. Abraham Silberschatz, Henry Korth, S. Sudarshan : Database Systems Concepts, TMH
3. Date/Kanna : An Introduction to Database Systems, ISBN, 9788177585568 , Pearson
4. Elmasri : Fundamentals of Database Systems, ISBN:9788131716250 , Pearson
5. Kristina Chodrow, MongoDB : the definitive guide
6. KorryDouglas, PostgreSQL, , ISBN:9780672327568
7. John Worsley, Joshua Drake Practical PostgreSQL (B/CD), ISBN: 9788173663925 Shroff / O'reilly
8. Joshua D. Drake, John C Worsley Practical Postgresql, O'Reilly
9. Richard Stones, Neil Matthew Beginning Databases with PostgreSQL, From Novice to Professional, 2nd Edition

https://www.javatpoint.com/mongodb-tutorial
https://www.tutorialsteacher.com/mongodb

Additional References:

Online course from Great Learning Free Courses on Database Management System

****

IMT- 415: Practical course on Object Oriented Concepts using C++ and Database Management System [4 Credits]

Course Objectives:

- The students should able to explain fundamental properties of the C++ language.
- The student should be able demonstrate, compile, debug &write programme.
- To solve real world computational problems.
- To perform operations on relational database management systems.
- Design E-R Model for given requirements and convert the same into database tables.
- Understand oops concepts in real life

Course Outcomes:

- Explain object oriented concepts and describe how they supported by C++
- Analyse , design and build object oriented software
- To use SQL & PL/SQL.
- To perform advanced database operations.
- Create database tables in PostgreSQL.
- Analyse and apply the class concepts in programming design.

Operating Environment:
For Object Oriented Concepts of C++:

- Operating system: Linux
- Editor: Any linux based editor like vi, gedit etc.
- Compiler : cc or gcc
- PostgreSQL : for DBMS

Programme of Object-Oriented Programming in C++
Assignment 1 - Simple programmes
Assignment 2- Decision Making
Assignment 3 - loop Control Structures.
Assignment 4 – programme of Function
Assignment 5- programme on Array
Assignment 6 – Class & objects
Assignment 7 – Inheritance
Assignment 8 – Function Overloading& Overriding
Assignment 9 – Friend function
Assignment 10 - File handling

Programme of DBMS

Assignment 1 - Draw E-R Diagrams
Assignment 2 - Create Tables
Assignment 3 – Create Tables using constraints, keys
Assignment 4- Create table and Relationships
Assignment 5 – Simple Queries
Assignment 6 – Nested Queries
Assignment 7 – Stored Procedure &Functions
Assignment 8 – Cursor & Trigger
Assignment 9 – Views
Assignment 10 - Case Study

****

IMT- 416(A): Statistical Methods [2T+2P=4 Credits]

Course Objectives:

- To use open source statistical software for data analysis.
- To understand the difference between univariate, bivariate and multivariate data and different measures used for analysing these types of data.
- To analyse univariate data using summary statistics tools.
- To analyse of bivariate data by using appropriate statistical techniques like correlation, regression and time series analysis.
- To analyse multivariate data by using multiple linear regression model
- To simulate the data from standard probability models.
- To fit the appropriate standard probability models to the given data

Course Outcomes:

- Understand the statistical tools and techniques for data condensation, data presentation and data analysis.
- Understand the discrete and continuous probability models.
- Distinguish between univariate, bivariate and multivariate data
- Apply the appropriate statistical methods for analysing different types of real data sets and interpret it.
- Simulate a random sample from discrete and continuous probability distributions
- Fit a discrete and continuous probability model to a given data
- Use open source statistical software for data analysis.

Course Content

Unit 1. Study of Open Source Statistical software [5 Hours]

Open source software: RStudio, PSPP, JASP, and jamovi

Unit 2. Data Condensation and Data Presentation [10 Hours]

2.1 Introduction, Definition of Statistics, Applications of Statistics in Mathematics and computer science,
2.2 Types of Data: Attributes- and Variables-Discrete and continuous variables, concept of univariate, bivariate and multivariate data
2.3 Categorical data and scale data
2.4 Primary Data and Secondary Data
2.5 Concept and definition of Population, target population, statistical population and sample
2.6 Methods of data collection: Census method and sampling method
2.7 Random Sampling methods: SRSWR, SRSWOR, Stratified random sampling, Systematic sampling, Cluster sampling, multistage sampling.
2.8 Tabulation and frequency Distributions using R-software
2.9 Graphical representation: Histogram, Frequency curve and Ogive curves using RStudio

Unit 3. Summary statistics for Univariate data [10 Hours]

3.1 Measures of Central tendency: Calculation of Mean, Mode and Median for ungrouped and grouped data and their interpretations
3.2 Measures of Dispersion: Calculation of Absolute measures- Range, Q.D., Variance and S.D. for ungrouped and grouped data and their interpretations.
3.3 Calculation of Relative measures- Coefficient of range, Coefficient of Q.D., Coefficient of variation for ungrouped and grouped data and their interpretations.
3.4 Concept and types of Skewness and Kurtosis, interpretation of skewness based on relation among 1) mean, median and mode 2) quartiles. Boxplot and its interpretation.
3.5 Examples and problems
3.6 Calculation of summary statistics for ungrouped data using PSPP

Unit 4. Summary statistics for Univariate data [20 Hours]

4.1 Correlation: Concept and applications of correlation. Types of correlation with examples. Measures of correlation: Scatter diagram, Karl Pearson's measure of correlation, Spearman's rank correlation and their interpretation.
4.2 Linear Regression models: Y on X line and X on Y line. Regression coefficients their properties and interpretation. (only the equations of lines are required), coefficient of determination and its significance in linear regression models

4.3 Non-linear Regression Models: Parabolic curve, Exponential curve and logistic curve, Multiple R and its significance in non-linear regression models

4.4 Time Series Models: Concept, definition and applications of Time series, Components of time series, Additive and Multiplicative models of Time series Measures of trend: Methods of moving averages, method of least squares, and method of exponential smoothing. AR(1) and AR(2) models

4.5 Multiple linear regression models: Concept and applications of multiple linear regression models. Equations of multiple linear regression models for trivariate data on (X1, X2, X3), partial regression coefficients and their interpretation, Multiple and partial correlation coefficients: Concept, definition, limits and their significance in multiple linear regression models.

4.6 Examples and problems

4.7 Use Open source statistical software PSPP and jamovi for regression modelling.

Unit 5. Standard Probability Models [15 Hours]

5.1 Concept of discrete and continuous random variables (r.v.). Definitions of p.m.f of discrete r.v., p.d.f. of continuous r.v., Expectation and variance of discrete and continuous r.v

5.2 Discrete probability Models: Binomial (n, p), Poisson (λ) and Geometric(p) distributions (definition, mean, variance, calculation of probabilities, applications, recurrence relation for probabilities, simulation and fitting of distributions)

5.3 Continuous probability Models: Uniform (a, b), Exponential (θ) and Normal (μ, σ2) distributions (definition, mean, variance, distribution function, calculation of probabilities, applications, simulation and fitting of distributions)

5.4 Use of RStudio and JASP for probability models

Reference Books:

1. Statistical Methods, George W. Snedecor, William G, Cochran, John Wiley &sons
2. Modern Elementary Statistics, Freund J.E. 2005, Pearson Publication
3. Fundamentals of Applied Statistics (3rd Edition), Gupta and Kapoor, S. Chand and Sons, New Delhi, 1987.
4. Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). The World Press Pvt. Ltd., Calcutta
5. A First course in Probability, Sheldon Ross. Pearson Education Inkc.
6. Mathematical Statistics (3rd Edition), Mukhopadhyay P. 2015, Books And Allied (P), Ltd.

Additional References:

1. Statistics using R : Narosa Publishing house by Dr. S. G. Purohit, Dr. S. D. Gore, Dr. S. R. Deshmukh
2. R for Data Science :Hadley Wickham and Garrett Grolemund : O'Reilly Publications

****

IMT- 416(B): Numerical Analysis [(2T+2P) =4Credits]

Course Objectives:

- An ability to apply knowledge of mathematics and computer science in practice.
- An ability to identify, critically analyze, formulate and solve problems with comprehensive knowledge in the area of specialization
- To improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.

Course Outcomes:

- Apply appropriate numerical methods to solve the problem with most accuracy. Apply the methods to solve linear and nonlinear equations.
- Find numerical integration and analyze error in computation.
- Solve differential equations using various numerical methods.
- Determine Eigen values and Eigen vectors of a square matrix.
- Implement numerical methods for a variety of multidisciplinary applications and a variety of numerical algorithms using appropriate technology.
- Compare different methods in numerical analysis with accuracy and efficiency of solution.

Course Content

Unit 1. Transcendental and Polynomial Equations [12 Hours]

1.1 Newton Raphson method
1.2 Regula Falsi method
1.3 Secant method
1.4 Fixed-Point iteration
1.5 Rate of convergence (secant method & Newton Raphson method)
1.6 Birge-Vieta method
1.7 Bairstow method

Unit 2. Systems of Linear equations [12 Hours]

2.1 Gauss elimination method
2.2 Triangularization method
2.3 Matrix factorization methods (Doolittle reduction, Crout reduction).
2.4 LU Decomposition method
2.5 Partition method
2.6 Iterative method for A^{-1}
2.7 Gauss-Seidel iteration

Unit 3. Polynomial Interpolation [10 Hours]

3.1 Finite difference operators
3.2 The Lagrange interpolation polynomial
3.3 Divided difference interpolation
3.4 Aitken's Algorithm
3.5 Choice of nodes and non-convergence of polynomial interpolation

Unit 4. Differentiation and Integration [10 Hours]

4.1 Numerical differentiation
4.2 Numerical integration
4.3 Double integration (Trapezoidal and Simpson's method)
4.4 Newton-Cotes methods
4.5 Error estimates for trapezoidal rule and Simpson's rule

Unit 5. Numerical solution of Differential Equations [10 Hours]

5.1 Euler's method
5.2 Analysis of Euler's method
5.3 Order of Euler's metho
5.4 Runge-Kutta method
5.5 One step modified and midpoint methods
5.6 Runge-Kutta methods for systems of equations.

Unit 6. The Eigen value problem [6 Hours]

6.1 Power method
6.2 Eigen values of symmetric matrices
6.3 Jacobi method

Reference Books:

1. M. K. Jain, S. R. K. Iyengar, R. K. Jain, Numerical methods for scientific and Engineering Computation (Fifth Edition), New Age International Publishers 2007.
2. John H. Mathews: Numerical Methods for Mathematics, Science and Engineering (Prentice Hall) 2nd Edition.

Additional References:

1. S. S. Sastry, Introductory methods of Numerical Analysis (Fifth Edition), PHI learning Private Limited, New Delhi 2012.
2. K. E. Atkinson: An introduction to Numerical Analysis (John Wiley Sons).

****

IMT- 416(C): Operations Research
[(2T+2P)=4Credits]

Course Objectives:

- Students will learn to identify the concept of Linear Programming Problem.
- Students will learn to apply methods to solve Integer programming problems and analysis of the solutions.
- Students will learn to understand the effect of variations in input data through sensitivity analysis.
- Students will learn to analyze the primal-dual relationship of a linear programming problem and compute the dual.
- Students will learn to understand techniques and methods to solve Transportation problems and Assignment problems.
- Students will learn to understand the concept of Non-linear programming and methods of solving the Non-linear programming problems.

Course Outcomes:

- Formulate real life problems into linear programming problem.
- Understand the importance of sensitivity analysis in managerial decision making.
- Analyze the effect of variations in input data of linear programming problem through sensitivity analysis.
- Understand the importance of duality in linear programming problem.
- Understand transportation model and will be able to find Initial Basic Feasible Solution and optimal transportation cost.
- Understand assignment models.
- Use the quadratic programming models for real life problems.

Course Content

Unit 1. Linear Programming Problem (LPP) [20 Hours]

1.1 Standard form of LPP
1.2 Simplex method
1.3 Big-M method
1.4 Types of linear programming solutions
1.5 Duality in LPP
1.6 Primal-Dual Relationship
1.7 Economic interpretation of Dual variables and constraints
1.8 Managerial significance of duality
1.9 Solution of Primal LPP using dual LPP
1.10 Practical on Simplex method
1.11 Practical on Duality in LPP
1.12 Practical on LPP by using Graphical method
1.13 Practical on Solutions of LPP using Excel solver.

Unit 2. Sensitivity Analysis in Linear Programming [16 Hours]

2.1 Changes in objective function coefficients

2.2 Changes in availability and resources

2.3 Changes in the input-output coefficients

2.4 Practical on Sensitivity analysis by using Excel solver

Unit 3. Integer Linear Programming [17 Hours]

3.1 Types of Integer Programming Problems

3.2 Gomory's all integer cutting plane method

3.3 Gomory's mixed- integer cutting plane method

3.4 Branch and Bound Method

3.5 Practical on Integer programming problems by using Excel solver

Unit 4. Transportation Model and Its Variants [20 Hours]

4.1 Transportation Model

a) Definition of Transportation Problem

b) Types of Transportation problem - unbalanced and balanced

4.2 Initial Basic feasible solutions

a) North West corner method

b) Least cost method

c) VAM method

4.3 Optimum Solutions by MODI method

a) Alternate optimal solution case

b) Unique optimal solution case

c) Prohibited transportation route problems

d) Maximization transportation problems

4.4 The Assignment Model and its solutions

a) Minimization Hungarian method

b) Multiple optimal solutions

c) Maximization case in assignment problems

d) Unbalanced assignment problem

e) Restrictions on assignment case

4.5 Traveling Salesperson Problem

4.6 Practical on solutions of Transportation problems by Excel Solver

4.7 Practical on Assignment model

4.8 Practical on Traveling Salesperson Problem by Excel Solver

Unit 5. Non-Linear Programming Problem [17 Hours]

5.1 The general non-linear programming problem

5.2 Graphical solution method

5.3 Quadratic Programming

a) Kuhn-Tucker Conditions

b) Wolfe's Modified Simplex Method

c) Beale's Method

5.4 Practical on Non-linear programming problem by Graphical Method

5.5 Practical on Quadratic Programming using Excel Solver

Recommended Book:

1. J. K. Sharma, Operations Research, (Third Edition, Macmillan India Ltd.), 2008.

2. Hamdy A. Taha, Operations Research, (Eighth Edition, Prentice Hall of India), 2008.

Reference Books:

1. P. K. Gupta and D. S. Hira, Operations Research, (Fifth Edition, S. Chand), 2014.

****

IMT-417: Research Methodology [2(T) +2(P) =04 Credits]

Course Description:

The Research Methodology course is designed to equip students in Mathematics with the essential skills and knowledge required to conduct rigorous and effective research in their field. This course provides an overview of various research methods, techniques, and tools commonly used in mathematical research, with an emphasis on developing critical thinking, problem-solving abilities, and research ethics. Students will also gain hands-on experience in formulating research questions, designing experiments, analysing data, and presenting and writing research findings.

Course Objectives:

- To develop a comprehensive understanding of different research methodologies and their applications in mathematics.
- To cultivate critical thinking and analytical skills necessary for identifying research problems and formulating research questions.
- To provide practical experience in designing experiments, collecting and analyzing data, and interpreting research results.
- To foster effective communication skills for presenting research findings orally and in written form.
- To promote ethical research practices and awareness of responsible conduct in mathematical research.

Course Duration:

This course is typically spread over one semester, equivalent to approximately 15 weeks of instruction.

Course Outline:

Foundations of Research:

Meaning, Objectives, Motivation, Utility, Concept of theory, Research Problem Identification, Developing a Research Plan – Exploration, Description, Diagnosis, Experimentation, Determining Experimental and Sample Designs. Writing of Proofs, quantifiers etc.

Research Design:

Defining research objectives and questions, Analysis of Literature Review – Primary and Secondary Sources, Web sources for critical Literature Review such as MathSciNet, ZMATH, Scopus, Web of Science, Reviewing literature and identifying research gaps.

Research Methods:

Scientific methods, Logical Methods: Deductive, Inductive, logical methods. Quantitative research methods, Qualitative research methods, Data Collection Techniques, Surveys and questionnaires, Interviews and focus groups, Observations and case studies, Experimental methods, Data Analysis and Interpretation, Statistical analysis techniques in mathematics, Qualitative data analysis methods, Visualization and interpretation of results.

Research Writing and Presentation:

Scientific/ technical Writing Structure and Components, Importance of Effective Communication. Preparing Research papers for journals, Seminars and Conferences – Design of paper using TEMPLATE, Calculations of Impact factor of a journal, citation Index, ISBN & ISSN. Preparation of Project Proposal – Time frame and work plan – Budget and Justification – Preparation and Publication of Research paper, Thesis writing. Project Reports for various funding, Writing Statement of Purpose for PhD/Post Doc etc, Writing a review of paper, Presenting research findings orally and visually, Research Collaboration and Communication, Collaborative research practices, Effective communication in mathematical research, Participating in conferences and seminars,

Research Ethics and Responsible Conduct:

Ethics and Ethical Issues – Ethical Committees – Commercialization – copy right – royalty – Intellectual Property rights and patent law – Track Related aspects of intellectual property Rights – Reproduction of published material – Plagiarism and software to detect plagiarism– Citation and Acknowledgement – Reproducibility and accountability.

Mathematical Software and Paraphrasing Software:

Basic Latex, Beamer, Overleaf, Grammarly, QuillBot, ChatGPT, and SAGE. Particularly, introduction to SAGE: Overview of the SAGE software, installation, and user interface. Basic Algebraic Manipulations: Symbolic algebra, equations, simplifications, and algebraic manipulations. Calculus Computations: Differentiation, integration. Linear Algebra with SAGE: Matrix operations, solving linear systems, eigenvalue calculations. Discrete Mathematics with SAGE: Combinatorics, graph theory, number theory, and cryptography.

Course Assessment:

The course assessment will be done at the college/institute that includes but is not limited to a combination of the following methods:

- Research proposals and progress reports
- Research presentations
- Critical analysis of published mathematical research papers
- Participation in class discussions and activities
- Final research project or paper

Note: The syllabus provided above is a general outline and can be adapted and expanded based on the specific requirements of the institution offering this subject in Mathematics programme and the expertise of the instructor.

References:

- Kothari, C.R.(2008), Research Methodology: Methods and Techniques. Second Edition. New Age International Publishers, New Delhi.
- Dilip Datta, LaTeX in 24 Hours, A Practical Guide for Scientific Writing, Springer
- Eva O. L. Lantsoght, The A-Z of the PhD Trajectory -A Practical Guide for a Successful Journey, Springer Cham, 2018.

****

Semester-II

IMT- 421: Foundation of Analysis [2(T) +2(P) =04 Credits]

Course Objectives:

- Understand several standard concepts of metric spaces and their properties like openness, closeness, completeness, Bolzano-Weierstrass property, compactness, and connectedness.
- Identify the continuity of a function defined on metric spaces and homeomorphisms.
- Understand many properties of the real line R and learn to define sequence in terms of functions from R to a subset of R.
- Recognize bounded, convergent, divergent, Cauchy and monotonic sequences and to calculate their limit superior, limit inferior, and the limit of a bounded sequence.
- Apply the ratio, root, alternating series and limit comparison tests for convergence and absolute convergence of an infinite series of real numbers.
- Learn some of the properties of Riemann integrable functions, and the applications of the fundamental theorems of integration.

Course Outcomes:

- Understand several standard concepts of metric spaces and their properties like openness, closednes, completeness, Bolzano-Weierstrass property, compactness,
- and connectedness.
- Identify the continuity of a function defined on metric spaces and homeomorphisms.
- Understand many properties of the real line R and learn to define sequence in terms of functions from R to a subset of R.
- Recognize bounded, convergent, divergent, Cauchy and monotonic sequences and to calculate their limit superior, limit inferior, and the limit of a bounded sequence.
- Apply the ratio, root, alternating series and limit comparison tests for convergence and absolute convergence of an infinite series of real numbers
- Learn some of the properties of Riemann integrable functions, and the applications of the fundamental theorems of integration.

Course Content

Unit 1. Metric Spaces and its Topology **[15 Hours]**

1.1 Metric Spaces Definition and Examples and k-cells.
1.2 convex sets, open closed ball and properties.
1.3 Neighborhood, limit point, isolated points, closed sets, interior points, open sets, perfect sets, bounded sets.
1.4 Dense sets, examples and properties.
1.5 Open cover, compact sets, examples and properties.
1.6 Connected sets, definition of separated sets, connected sets and properties.

Unit 2. Numerical Sequences and series **[15 Hours]**

2.1 Convergent Sequences, Definition and Examples Properties, Definition and properties.

2.2 Cauchy Sequences: Definition, Examples and properties, Definition of complete. metric space, examples, Definition of Monotonic Sequences and its properties.

2.3 Upper and lower limits, Definition, examples and properties.

2.4 Convergence of some special sequences.

2.5 Series: Definition, examples and properties, series of non-negative terms, Cauchy's condensation test and examples.

2.6 The Number e.

2.7 Root and ratio tests, examples

2.8 Power series, Definition, radius of Convergence, examples and properties

2.9 Summation by parts, absolute convergence

Unit 3. Continuity **[16 Hours]**

3.1 Limits of functions: Definition, examples and properties.

3.2 Continuous functions, Definition, examples and properties.

3.3 Continuity and Compactness.

3.4 Bounded Set: Definition.

3.5 Continuous image of a compact set is compact and related properties.

3.6 Definition of Uniform Continuity and related properties.

3.7 Continuity and Connectedness: continuous image of connected set is connected and related properties.

3.8 Discontinuities, Definition, examples.

3.9 Monotonic functions, Definition examples and properties.

Unit 4. Differentiation **[10 Hours]**

4.1 Derivative of a real function, Definition examples and properties.

4.2 Mean Value Theorem.

4.3 Continuity of derivatives.

4.4 Taylor's theorem.

4.5 Differentiation of a vector valued function.

Unit 5. Riemann Stieljes Integral **[19 Hours]**

5.1 Definition and existence of the integral, related properties.

5.2 Properties of the integral.

5.3 Integration and differentiation.

5.4 Integration of vector valued functions.

Unit 6. Sequences and series of function **[15 Hours]**

6.1 Discussion of main problem- with examples

6.2 Uniform convergence: Definition and properties

6.3 Uniform convergence: and continuity

6.4 Uniform convergence: and integration

6.5 Uniform convergence: and differentiation

Reference Books:

1. Walter Rudin: Principles of Real Analysis, (3rd Edition, Tata McGraw Hill Publication) Art. 2.15 to 2.42, 2.45 to 2.47, Art. 3.1 to 3.46, Art. 4.1 to 4.18 4.19 (Statement only), 4.22 to 4.28, 4.29 (Statement only), 5.1 to 5.12, 5.15 to 5.19, 6.1 to 6.15, 6.20, to 6.25, Art 7.1 to 7.17.
2. Ajit Kumar and S. Kumaresan A Basic Course in Real Analysis.

Additional References:

1. C. C. Pugh, Real Mathematical Analysis.
2. T. M. Apostol, Mathematical Analysis
3. G. F. Simmons, Topology and Modern Analysis

****

IM 422: Applied Algebra [4 Credits]

Course Objective:

- To learn the significance and relevance of abstract algebra.
- To learn basic examples of algebraic structures and the operations upon these structures.
- To learn necessary tools and techniques in algebra with a view towards applications of algebra.
- To learn the applications to cryptography.

Course Outcomes:

- To understand the notion of group and its examples with a special emphasis on group of integers modulo n.
- To learn the basics of modular arithmetic and computations.
- To learn basic techniques and algorithms in algebra.
- To learn some applications of algebra such as applications to Cryptography.

Course Content

Unit 1. Groups [16 Hours]

1.1 Definition and Examples of Groups
1.2 Symmetries of a square
1.3 Dihedral Groups
1.4 Elementary Properties of Groups
1.5 Finite Groups
1.6 Subgroups, Subgroup Tests
1.7 Cyclic Group, Properties of Cyclic Groups
1.8 Classification of Subgroups of Cyclic Groups

Unit 2. Permutation Groups **[14 Hours]**

2.1 Permutation Groups and its properties
2.2 Isomorphism: Definition and examples
2.3 Properties of Isomorphisms
2.4 Cosets, Properties of Cosets
2.5 Lagrange's Theorem and Consequences

Unit 3. Number Theory **[30 Hours]**

3.1 Division Algorithm and GCD
3.2 Mathematical Induction
3.3 Primes and Uniqueness of Factorization
3.4 Congruences
3.5 Solving Linear Congruences
3.6 Euler's Theorem and Public Key Codes

Reference Books:

1. Contemporary Abstract Algebra By J. A, Gallian (Seventh Edition)
   Unit 1: Chapter 1 to 4
   Unit 2: Chapter 5 to 7
2. Numbers Groups and Codes by J. F. Humphreys

Additional References:

1. Bernard Kolman, Robert C. Busby and Sharon Ross, Discrete Mathematical Structures (6^{th} Edition) Pearson Education Publication
2. J. B. Fraleigh: A First Course in Abstract Algebra
3. I. Niven, H. Zuckerman and H.L. Montgomery, An Introduction to Theory of Numbers, 5th Edition, John Wiley and Sons.
4. David M. Burton, Elementary Number Theory (Second Ed.), Universal Book Stall, New Delhi, 1991.

****

IMT- 423: Data Structure [02 Credits]

Course Objectives:

- To learn various structures to represent data.
- To learn the efficient way of problem solving.
- To understand the different methods of organizing large amount of data.
- To efficiently implement the linear and non-linear data structure.

Course Outcomes:

- Implementation of different data structures efficiently.
- Usage of well-organized data structures to handle large amount of data.
- Usage of appropriate data structures for problem solving.

Course Content

Unit 1. Array as a Data Structure [4 Hours]

1.1 Data structure, types- linear and nonlinear data structure

1.2 Array introduction, need for array, representation, basic operation on array-create static and dynamic array, traverse, insertion, deletion.

1.3 Array applications–Searching method- Sequential search, Binary Search, Sorting, Method - Insertion sort,

1.4 Bubble sort, Merge Sort, Quick sort.

Unit 2. Linked List [5 Hours]

2.1 List as a Data Structure.

2.2 Types of Linked List–Singly, Doubly, Circular.

2.3 Operations on Linked List-create, traverse, insert, delete, search, sort, reverse.

2.4 Applications of Linked List– polynomial representation, Addition of two polynomials.

Unit 3. Stack [5 Hours]

3.1 Introduction of stack.

3.2 Operations – init(), push(), pop(), isEmpty(), isFull(), peek().

3.3 Applications of stack- Recursion, String reversal, palindrome checking, Expression types - infix, prefix and postfix, expression conversion and evaluation.

Unit 4: Queue [5 Hours]

4.1 Introduction of queue.

4.2 Operations - init(), enqueue(), dequeue(), is Empty(), is Full(), peek().

4.3 Types of Queues-Linear Queue,Circular Queue, Priority Queue, Double Ended.

4.4 Applications–CPU Scheduling (FCFS).

Unit 5. Tree [5 Hours]

5.1 Concept and Terminologies

5.2 Types of Binarytrees-Binarytree, skewedtree, strictly binary tree,full binary tree, complete binary tree, expression tree, binary search tree, Heap tree

5.3 Tree traversals– pre-order, in-order, post-order, Counting leaf, non-leaf,and total nodes

5.4 Terminology: Balanced trees-AVL Trees –concept and rotations.

5.5 Applications of trees- Heap Sort.

Unit 6: Programming Assignments **[6 Hours]**

Out of 30 lectures- 24 theory lectures and 6 are assigned for demonstration. Teacher should give demonstration of various programmes mentioned below using C++ in the classroom or in the laboratory as per their convenience.

Assignment 1: Implementation of searching method -sequential search and binary search.

Assignment 2 Implementation of sorting method- Bubble sort or merge sort.

Assignment 3 Implementation of singly or doubly linked list with operation- create, traverse, insert and delete.

Assignment 4 Implementation of stack with operation- create, traverse, push () and pop().

Assignment 5 Implementation of Linear Queue with operation- init (), enqueue (), dequeue (), isEmpty (), isFull ().

Assignment 6 Implementation and Operations on Binary Search Tree - Create, Insert, Delete.

Reference Books

1. Classic Data Structures-D.Samanta, Prentice Hall India Pvt.Ltd.
2. Fundamentals of Data Structures in C- Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, 2nd Edition, Universities Press.
3. Data Structures using C and C++-Yedidyah Langsam, MosheJ. Augenstein,Aaron M.Tenenbaum, Pearson Education
4. Data Structures:A Pseudocode approach with C,Richard Gilberg, Behrouz A.Forouzan, Cengage Learning.
5. Introduction to Data Structures in C-Ashok Kamthane, Pearson Education.
6. AlgorithmsandDataStructures,NiklausWirth,PearsonEducation.
7. https://www.tutorialspoint.com/data_structures_algorithms/index.htm

Additional References:

Free Online course as a part of assignment can be given to student. Link as follows:

1. https://www.udemy.com/course/data-structures-for-beginners-c-plusplus/
2. https://www.classcentral.com/course/freecodecamp-data-structures-full-course-using-c-and-c-57801

****

IMT-424: Java Programming (Lab Course) [04 Credits]

Course Objectives:

- To understand the concepts of object-oriented paradigm in the Java programming language.
- To understand the importance of Classes & objects along with constructors, Arrays, Strings.
- To learn the principles of inheritance, interface and packages and demonstrate the concept of reusability for faster development.
- To recognize usage of Exception Handling, Multithreading, Collection Framework, Java-Database Connectivity.
- To learn designing, implementing, testing, and debugging graphical user interfaces in Java using Swings and AWT components that can react to different user events.

Course Outcomes:

On successful completion, of course, learner/student will be able to:

- Explain the fundamental concepts of Java Programming.
- Use the concepts of classes, objects, members of a class and the relationships among them needed for finding the solution to specific problems.
- Demonstrate how to extend java classes and achieve reusability using Inheritance, Interface and Packages.
- Construct robust and faster programmed solutions to problems using the concept of Multithreading, exceptions and Collection Framework.
- Perform database connectivity with java and database tools like MySQL, PostgreSQL.
- Design and develop Graphical User Interface using Abstract Window Toolkit and Swings along with response to the events.

Course Content

Unit 1: Java Fundamentals **[6 Hours]**

1.1 Introduction to Java
1.2 Keywords, Data types, Variables, Operators, Expressions
1.3 Basic Java tools
1.4 Control Statements: If Statement, switch Statement, break, continue.
1.5 Iteration Statements: for loop, while loop, and do-while loop

Unit 2: Classes, objects, Arrays and Strings **[10 Hours]**

2.1 Defining Your Own Classes
2.2 Access Specifiers (public, protected, private, default)
2.3 Array of Objects
2.4 Constructor, Overloading Constructors and use of ‘this’ Keyword

2.5 Static keyword
2.6 Predefined class Object class methods
2.7 Inner class
2.8 Creating, Accessing and using Packages
2.9 Creating jar file and manifest file
2.10 Wrapper Classes
2.11 Garbage Collection (finalize() Method)
2.12 String, StringBuffer
2.13 Inbuilt String functions

Unit 3: Inheritance **[10 Hours]**

3.1 Inheritance Basics (extends Keyword) and Types of Inheritance
3.2 Superclass, Subclass and use of Super Keyword
3.3 Method Overriding and runtime polymorphism
3.4 Use of final keyword related to method and class
3.5 Use of abstract class and abstract methods
3.6 Defining and Implementing Interfaces
3.7 Runtime polymorphism using interface
3.8 Object Cloning

Unit 4: Exception Handling **[6 Hours]**

4.1 Exception-Handling Fundamentals
4.2 Exception Types
4.3 Exception class Hierarchy
4.4 Using try and catch
4.5 Multiple catch Clauses
4.6 Nested try Statements, throw, throws, finally
4.7 Java's Built-in Exceptions
4.8 Creating Your Own Exception Subclasses

Unit 5: Multithreading **[8 Hours]**

5.1 What are threads
5.2 Life cycle of thread
5.3 Running and starting thread using Thread class
5.4 Thread priorities
5.5 Running multiple threads
5.6 The Runnable interface
5.7 Synchronization and inter thread communication

Unit 6: Collection Framework **[8 Hours]**

6.1 Wrapper Classes
6.2 Introduction to the Collection framework
6.3 List – ArrayList, LinkedList and Vector
6.4 Set - HashSet, TreeSet, and LinkedHashSet
6.5 Map – HashMap, LinkedHashMap, Hashtable and TreeMap
6.6 Interfaces such as Iterators, ListIterators, Enumerations

Unit 7: GUI programming- I (AWT, Event Handling, Swing) **[8 Hours]**

7.1 What is AWT ? What is Swing? Difference between AWT and Swing.
7.2 The MVC Architecture and Swing
7.3 Layout Manager and Layouts, The JComponent class
7.4 Dialogs (Message, confirmation, input), JFileChooser, JColorChooser
7.5 Event Handling: Event sources, Listeners
7.6 Mouse and Keyboard Event Handling
7.7 Adapters
7.8 Anonymous inner class

Unit 8: Java Database Connectivity **[4 Hours]**

8.1 The design of JDBC
8.2 Basic JDBC programme Concept
8.3 Drivers
8.4 Making the Connection, Statement , ResultSet
8.5 Executing SQL commands
8.6 Executing queries
8.7 MetaData

List of Practical

1. Java Tools and IDE, Simple java programmes **[4 Hours]**

- Introduction to the java environment
- Use of java tools like java, javac, jdb and javadoc
- Simple Programme using Control structure and Looping statements

2. Methods, Classes and Objects **[4 Hours]**

- Defining simple methods,
- Using Recursion
- Defining simple classes and creating objects.

3. Arrays, String function **[4 Hours]**

- Defining Arrays, Traversing Arrays
- Searching and Sorting elements in Arrays
- Defining String, using various inbuilt string functions
- Implementing String, StringBuffer and StringBuilder.

4. Inheritance **[6 Hours]**

- Implement inheritance in java.
- Creating abstract classes.

- Defining and using interfaces.

5. Packages **[6 Hours]**

- Using predefined packages
- Creating packages,
- Creating subpackages

6. Exception Handling, **[4 Hours]**

- Demonstrate Exception Handling Mechanism in Java.
- Use of try, catch, throw, throws, finally blocks
- Defining User defined Exception classes.

7. Multithreading **[4 Hours]**

- Creating Thread
- Demonstrate thread life cycle and various states

8. Collection Framework **[10 Hours]**

- Add, retrieve & remove element from ArrayList
- Implement LinkedList
- Sort & reverse the LinkedList elements
- Implement push() and pop() on Stack
- Implement binary search.

9. AWT and Swing **[10 Hours]**

- To demonstrate GUI creation using Swing Package and Layout managers.
- To understand Event handling mechanism in Java.
- Using Event classes, Event Listeners and Adapters.

10. Java Database Connectivity **[10 Hours]**

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & modifying the resulting records.

References:

1. Core Java : Volume I – Fundamentals By: Horstmann, C. S/ Cornell, G. 8th ed Pearson
2. Core Java 2: Volume I – Fundamentals By: Horstmann, C. S/ Cornell, G. 7th ed Pearson 20
3. Data Structures, Algorithms, & Application In Java By: Sahni, Sartaj MGH
4. Database Programming With Jdbc & Java By: Reese, George 2nd ed Oreilly
5. Head First Java By: Sierra, K/ Bates, B. 2nd ed Oreilly
6. Inside Java By: Siyan, K. S/ Weaver, J. L. New Riders

****

IMT-425(A): Web Technology - Part I [2(T) +2(P)=04 Credits]

Course Objectives:

- To introduce students to modern web technologies.
- To learn and use server side programming using Node.js
- To understand asynchronous programming.
- Learn Web Application Development using library Express.js

Course Outcomes:

- Students will be empowered to use technology that is widely used as part of full stack development
- Students will gain sufficient knowledge to develop Web Platforms which support Mobile Applications, Web Applications and other data consumers using Python or any other technology stacks
- Students will understand what really the asynchronous and event based programming techniques

Course Content

Unit 1 HTML5 [4 Hours]

1.1 Introduction to HTML5: Overview of HTML and its purpose, Understanding the structure of an HTML document, Setting up an HTML file using a text editor

1.2 Document Structure and Essential Tags: Basic structure of an HTML5 document which help create headings, paragraphs, links, and render images on the page.

1.3 HTML5 Forms: Creating forms using the form element, Text input fields, text area, checkboxes, and radio buttons, Using the label element for form controls, Submitting forms and handling user input

1.4 HTML5 Semantic Elements: Introduction to semantic elements in HTML5, Using header, nav, main, article, section, aside, footer, etc., Benefits of using semantic elements for accessibility and SEO

1.5 HTML5 Media Elements: Adding audio and video to the webpage using audio and video elements, Providing fallback content for non-supported browsers, Using the source element for multiple media formats

1.6 HTML5 Links and Navigation: Creating hyperlinks with <a> element, Understanding absolute and relative URLs, Navigating within a webpage using fragment identifiers (#) and anchor tags

Unit 2. Introduction to JavaScript [7 Hours]

2.1 JavaScript data types

2.2 Variables, Functions, Events, Regular Expressions

2.3 Array and Objects, JSON in Java Script

2.4 JavaScript HTML DOM
2.5 Promises and Callbacks

Unit 3. Introduction to Nodejs [3 Hours]

3.1 Introduction to Node
3.2. Node JS Process model
3.3. Installation of Node JS
3.4 Creating Web Server
3.5 Introduction to HTTP Protocol and its statelessness
3.6 Handling HTTP requests
3.7 Node JS Modules- functions, local and global module

Unit 4. Node Package Manager and File system [4 Hours]

4.1 What is NPM?
4.2 Installing package locally
4.3 Adding dependencies in package.json
4.4 Installing packages globally
4.5 Updating packages
4.6 Managing Dependencies
4.7 FS Modules: Files and Directories, Streams, Reading and Writing Files

Unit 5. Events [4 Hours]

5.1 Asynchronous JS
5.2 Asynchronous control flow with callbacks
5.3 Promises
5.4 EventEmitter Class
5.5 ASync, Await
5.6 Returning Event Emitter
5.7 Inheriting Events

Unit 6. Working with Databases with Node.js [4 Hours]

6.1 Introduction to databases (MongoDB)
6.2 Connection String
6.3 Configuring
6.4 Working with Select command
6.5 Various database operations
6.6 Mongoose ODM
6.7 Mongoose Schema
6.8 Mongoose Model
6.9 Querying with Mongoose

Unit 7 Introduction to Express.js [4 Hours]

7.1 REST API - Introduction and consuming it in Application
7.2 Introduction to Express JS
7.3 Routing, Responding
7.4 Configuration
7.5 Views

7.6 Receiving Data
7.7 Error Handling

Reference Books:

1. HTML 5 Black Book : Covers Css3, Javascript, XML, XHTML, Ajax, PHP And Jquery by Kogent Learning Solutions Inc, Published November 2011 by Dreamtech Press
2. Node.js complete reference guid , velentin Bojinov, David Herron, Dioge Resende, packt Publishing Ltd
3. Mastering Nod.js By Sandro Pasquali , packt Publishing
4. Smashing Node.js, Java Script Everywhere , Guillermo Rauch, John wiley & Sons
5. Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown.

****

IMT- 425(A): Web Technologies - Part II [(2T+2P=04 Credits]

Course Objectives:

- To introduce students to modern web technology concepts
- To learn and use server side programming using Node.js
- To understand asynchronous programming.
- To learn and understand web application using Express.js

Course Outcomes:

On completion of the course, student will be able to–

- The students will gain sufficient knowledge to use the latest technology trends to develop Web Technology platforms.
- Students will know the powerful way to develop the web application in Python and similar Programming languages.
- They will be able to use event based programming techniques as well as asynchronous programming into Application Development.

Topics of the Assignments

Assignment1: Practical based on HTML5 [8 Hours]
Assignment2: Practical based on JavaScript [12 Hours]
Assignment3: Practical based on Node JS [8 Hours]
Assignment4: Practical based on Node Package Manager and File system [8 Hours]
Assignment5: Practical based on Events [8 Hours]
Assignment6: Practical based on Databases using Node.js [8 Hours]
Assignment7: Build a system using Express.js [8 Hours]

****

IMT- 425(B): Financial Mathematics [(2T+2P=04 Credits]

Course Objective:

- Identify basic terminologies in Mathematical Finance
- State the concepts of Risk Free and Risky Assets
- Differentiate between forward and Futures
- Analyse the concept of Risk and apply it to build portfolio from various securities
- Describe the principle of No Arbitrage and Fundamental Theorem of Asset Pricing
- Design a scenario for evaluating American and European Options and determine its value over time

Course Outcomes:

- Students will be able to differentiate between risk and risk free assets
- Students will be able to construct and evaluate a portfolio with various investments
- Students will learn a basic terminologies with forward and future trading
- Students will learn basic terminologies in Option Trading and can Time Value the Options

Course Content:

Unit 1. Risk free Assets **[10 Hours]**

1.1 Time Value of Money- Simple Interest, Periodic Compounding, Stream of Payments, Continuous 12 Compounding, How to compare Compounding Methods, Money Market- Zero Coupon Bonds, 1.3 Coupon Bonds, Money Market Account.

Unit 2. Risky Assets **[10 Hours]**

2.1 Dynamics of Stock Prices- Returns, Expected Returns,
2.2 Binomial Tree Model- Risk Neutral Probability, Martingale Property
2.3 Other Models- Trinomial Tree Model, Continuous Time Limit.

Unit 3. Discrete Time Market Models **[10 Hours]**

3.1 Stock and Money Market Models- Investment Strategies,
3.2 Principle of no Arbitrage, Application to Binomial Tree Model
3.3 Fundamental Theorem of Asset Pricing, Extended Models

Unit 4. Portfolio Management **[10 Hours]**

4.1 Concept of Risk, Two Securities- Risk and Expected Return on Portfolio,
4.2 Several Securities- Risk and Expected Return on Portfolio, Efficient Frontier

4.3 Capital Asset Pricing Model, Beta Factor, Security Market Line,

Unit 5. Forward and Future **[08 Hours]**

5.1 Forward Contracts, Forward Price,

5.2 Value of a Forward Contract Futures-Pricing,

5.3 Hedging with Futures

Unit 6. Options-General Properties **[12 Hours]**

6.1 Definitions, Introduction to Put Call Parity Formula

6.2 Bounds on Option Prices- European Options, European and American Calls on Non Dividend

6.3 Paying Stock, American Options , Variables determining Option Prices- European and American Options

6.4 Time Value of Options.

Reference Book:

1. Mathematics for Finance: An introduction to Financial Engineering, Marek Capinski, Tomasz Zastawniak, Springer Publications

Additional Reference:

1 The Calculus of Finance, Amber Habib, Universities Press Investment Science, David Luenberger, Oxford University Press

****

IMT- 425(C): Computational Geometry [(2T+2P=04 Credits]

Course Objectives:

- To provide a review of transformations of the plane, including translations, reflections, rotations, shears, and their applications.
- To introduce homogeneous coordinates and their use in projective geometry and transformations. Further, to study projections, including parallel projection and perspective projection, and their types.
- To explore curve rendering techniques and the parametric representation of curves.
- To classify conics and understand their intersections with lines.
- To study Bezier curves of various degrees, including linear, quadratic, cubic, and general Bezier curves, and their properties.
- To introduce rational Bezier curves and their applications. Further, to explore B-splines, their properties, and their types, with applications in font design.
- To understand and analyze algorithms used in computational geometry, including the closest pair problem, collision detection, convex hull algorithms (Graham Scan, Gift Wrapping, Chan's),

smoothing algorithms, line segment intersection algorithms, nesting algorithm, point location with respect to a polygon, triangulation, and bounding box algorithms.

Course Outcomes:

- Students can able to use various transformations to solve the problems in various disciplines where computational geometry plays a vital role.
- Students can able to work on and able to construct various curves as per the requirement using various tools in this course.
- By using algorithms students can able to do various projects,
- With the help of some technical knowledge (programming elements) and computational geometry knowledge, students can able to enter in Geometry based companies.

Course Content

Unit 1. Revision **[10 Hours]**

1.1 Transformations of the Plane
1.2 Translations, reflections, rotations, shears, concatenation of transformations, applications
1.3 Homogenous coordinates: Homogenous coordinates, points at infinity, projective plan transformations in homogenous coordinates.
1.4 Transformations of the Space: Translations, scaling, reflection, rotation about coordinate axes, rotation about an arbitrary line, reflection in an arbitrary plane, applications to Computer-aided Design.

Unit 2. Projections **[6 Hours]**

2.1 Parallel projection and its types
2.2 Perspective projection and its types

Unit 3. Curves **[14 Hours]**

3.1 Curve rendering, parametric Curves, arc length and reparameterization
3.2 Classification of Conics, Intersections of a Conic with a Line, parametrization of an irreducible conic
3.3 Conics in space, applications of conics

Unit 4. Bezier Curves **[12 Hours]**

4.1 Bezier curves of low degree, linear Bezier curves, quadratic Beziercurves, cubic Bezier curves, the general Bezier curve
4.2 Properties of the Bernstein polynomials, properties of Bezier curves

4.3 The de Casteljau Algorithm and applications, Rational Bezier Curves and its properties and applications.

Unit 5. B-splines **[6 Hours]**

5.1 Introduction to B-splines,
5.2 Properties of the B-spline Curve and its types
5.3 Application to Font Design.

Unit 6. Algorithms **[12 Hours]**

6.1 Closest pair problem, Collision detection
6.2 Convex hull algorithms (Graham Scan, Gift Wrapping, Chan's)
6.3 Smoothing algorithms, Line segment intersection algorithms
6.4 Nesting algorithm, Position of a point with respect to polygon
6.5 Triangulation, Bounding box algorithm.

Reference Book:

1. Duncan Marsh, Applied Geometry for Computer Graphics and CAD (Springer, Second Edition) (Chapters 1, 2, 3, 4, 5, 6, 7, 8 (Section 8.1))

Additional Reference:

1. de Berg, van Kreveld, Overmars, and Schwarzkopf, Computational Geometry Algorithms and Applications, 2nd Edition, (Springer-Verlag, 2000).

****

IMT- 426: On Job Training(OJT) / Field Project [04 Credits = 120 Hrs]

In this course, the students are expected to do the On Job Training (OJT) in appropriate Industries/Government sectors/Institute etc. to get hands on experience. The department may conduct necessary lectures/workshops/seminars as a prerequisite for OJT. The course will be conducted as per the guidelines of the Department/the University and Government of Maharashtra.

****