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).

ProgrammeOutcomes (POs)

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

PO-No.	ProgrammeOutcomes *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.	**ProgrammeSpecificOutcomes** *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 forvarious 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 StructureoftheProgramme:M.Sc.(IMCA)Part- I

Approved by B.O.S.

Year	Level	Sem.	Course Type	CourseCode	CourseTitle	Remark	Credit	No. ofHours
1	6.0	I	Core	IMT-501MJ	LinearAlgebra	Theory	2	30
			Core	IMT-502MJP	Practical based on LinearAlgebra	Practical	2	60
			Core	IMT-503MJ	Discrete Mathematical Structure	Theory	4	60
			Core	IMT-504MJ	Object Oriented Programming using C++	Theory	2	30
			Core	IMT-505MJ	Data Base Management System	Theory	2	30
			Core	IMT-506 MJP	Lab on C++ and DBMS	Practical	2	60
			Elective (Chooseany one)	IMT-507(A)MJ	Statistical Methods	Theory	2	30
				IMT-508(A)MJP	Practical based onStatistical Methods	Practical	2	60
				IMT-507(B)MJ	Numerical Analysis	Theory	2	30
				IMT-508(B)MJP	Practical based onNumerical Analysis	Practical	2	60
				IMT-507(C)MJ	Operations Research	Theory	2	30
				IMT-508(C)MJP	Practical based onOperations Research	Practical	2	60
			RM	IMT-509MNP	ResearchMethodology	Theory	2	30
				IMT-510MNP	Practical based onResearchMethodology	Practical	2	60
		II	Core	IMT-551MJ	Foundation of Analysis	Theory	4	60
			Core	IMT-552MJ	Applied Algebra	Theory	4	60
			Core	IMT-553MJ	Data Structure	Theory	2	30
			Core	IMT-554MJP	Java Programming	Practical	4	120
			Elective (Chooseanyone)	IMT-555(A)MJ	Web Technology	Theory	2	30
				IMT-556(A)MJP	Practical based onWeb Technology	Practical	2	60
				IMT-555(B)MJ	Financial Mathematics	Theory	2	30
				IMT-556(B)MJP	Practical based on Financial Mathematics	Practical	2	60
				IMT-555(C)MJ	ComputationalGeometry	Theory	2	30
				IMT-556(C)MJP	Practical based onComputationalGeometry	Practical	2	60
			OJT/FP	IMT-581	OnJobTraining OrFieldProject	Practical	4	120

Course StructureoftheProgramme: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-601MJ	Ordinary Differential Equations	Theory	**4**	60
			Core	IMT-602MJ	Computer Networks	Theory	**2**	30
			Core	IMT-603MJ	Design and Analysis of Algorithm	Theory	**4**	60
			Core	IMT-604MJP	Data Analysis Using Python	Practical	**4**	120
			Elective (Choose Anyone)	IMT-605(A)MJ	Data Mining	Theory	2	30
				IMT-606(A)MJP	Practical based on Data Mining	Practical	2	60
				IMT-605(B)MJ	Dot Net	Theory	2	30
				IMT-606(B)MJP	Practical based on Dot Net	Practical	2	60
				IMT-605(C)MJ	Statistical Inference	Theory	2	30
				IMT-606(C)MJP	Practical based on Statistical Inference	Practical	2	60
			RP	IMT-631	Research Project	Practical	**4**	120
		IV	Core	IMT-651	Software Engineering and Testing	Theory	**4**	60
			Core	IMT-652	Optimization Techniques	Theory	**4**	60
			Core	IMT-653	Artificial Intelligence and Machine Learning	Practical	**4**	120
			Elective (Choose anyone)	IMT-654(A)MJ	Go Programming	Theory	2	30
				IMT-655(A)MJP	Practical based on Go Programming	Practical	2	60
				IMT-654(B)MJ	DEFI Block chain Technology	Theory	2	30
				IMT-655(B)MJP	Practical based on DEFI Block chain Technology	Practical	2	60
				IMT-654(C)MJ	Graph Theory	Theory	2	30
				IMT-655(C)MJP	Practical based on Graph Theory	Practical	2	60
			RP	IMT-681	Research Project	Practical	**6**	180

Details of Syllabus:

Semester-I

IMT-501MJ & 502MJP: 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]**

5.1Minima

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

6.3Computationof 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-503MJ:Discrete Mathematical Structure [04 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 [10 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 [10 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 Operations on graphs.

2.5 Elementary Edge connectivity, vertex connectivity

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

Unit 3.General Counting Methods for arrangements and selections [10 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 [10 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.

Unit 5.Mathematical Induction and Recursion [10 Hours]

5.1 Mathematical induction
5.2 Strong induction and well-ordering
5.3 Recursive definitions and structural induction
5.4 Recursive algorithms

Unit 6.Fundamentals of Algorithms [5 Hours]

5.1 Introduction to searching and sorting algorithms
5.2 The growth of functions

Unit 6.Boolean Algebra [5 Hours]

5.1 Introduction to Boolean functions
5.2 Representing Boolean functions
5.3 Introduction to Logic Gates

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-504MJ: 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-whileJump 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-505MJ: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 SQLJoins

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-506 MJP: Practical course on Object Oriented Concepts using C++ and Database Management System[2 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-507(A)MJ & 508(A)MJP: 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-507(B)MJ & 508(B)MJP: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 EngineeringComputation (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 learningPrivateLimited, New Delhi 2012.
2. K. E. Atkinson: An introduction to Numerical Analysis (John Wiley Sons).

****

IMT-507(C)MJ & 508(C)MJP: 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-509MJ & 510MJP:ResearchMethodology[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.
- DilipDatta, 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-551MJ:Foundation of Analysis [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 **[10 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 **[10 Hours]**

2.1Convergent 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 **[9 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 **[7 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 **[14 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 **[10 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. KumaresanA 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

****

IMT –552MJ: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-553MJ:Data Structure[02 Credits]

Course Objectives:

- To learn various structures to represent data.
- Tolearntheefficientwayofproblem solving.
- Tounderstandthedifferentmethodsoforganizinglargeamountof data.
- Toefficientlyimplement the linear and non-lineardatastructure.

Course Outcomes:

- Implementationofdifferentdatastructuresefficiently.
- Usageofwell-organizeddatastructurestohandlelargeamountofdata.
- Usageofappropriatedatastructuresforproblemsolving.

Course Content

Unit 1.Arrayasa 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-createstatic and dynamic array, traverse, insertion, deletion.

1.3 Arrayapplications–Searching method- Sequential search, BinarySearch, Sorting, Method - Insertion sort,

1.4 Bubble sort, Merge Sort, Quick sort.

Unit 2. Linked List **[5 Hours]**

2.1 ListasaData Structure.

2.2 TypesofLinkedList–Singly, Doubly, Circular.

2.3 OperationsonLinkedList-create,traverse,insert, delete,search,sort,reverse.

2.4 ApplicationsofLinkedList– polynomial representation, Additionoftwopolynomials.

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(), isEmpty(), isFull(), peek().

4.3 TypesofQueues-LinearQueue,Circular Queue, Priority Queue,Double Ended.

4.4 Applications–CPUScheduling (FCFS).

Unit 5: Tree **[5 Hours]**

5.1 ConceptandTerminologies

5.2 TypesofBinarytrees-Binarytree,skewedtreestrictlybinarytree,fullbinarytree,completebinary tree, expression tree, binary search tree, Heap tree

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

5.4 Terminology:Balancedtrees-AVLTrees–conceptandrotations.

5.5 Applicationsoftrees- Heap Sort.

Unit 6: ProgrammingAssignments **[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 theirconvenience.

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

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

Assignment3 Implementation of singly or doubly linked list with operation- create, traverse, insert and delete.

Assignment4 Implementation of stack with operation- create,traverse, push () and pop().

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

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

Reference Books

1. ClassicDataStructures-D.Samanta,PrenticeHallIndiaPvt.Ltd.
2. Fundamentals of Data Structures in C- Ellis Horowitz, SartajSahni, Susan Anderson-Freed,2nd Edition, Universities Press.
3. DataStructuresusingCandC++-YedidyahLangsam,MosheJ. Augenstein,Aaron M. Tenenbaum,Pearson Education
4. DataStructures:APseudocodeapproachwithC,RichardGilberg, BehrouzA.Forouzan, CengageLearning.
5. IntroductiontoDataStructuresin C-AshokKamthane,PearsonEducation.
6. AlgorithmsandDataStructures,NiklausWith,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-554MJP: 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
1.2Keywords, 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 **[8 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 edOreilly
5. Head First Java By: Sierra, K/ Bates, B. 2nd edOreilly
6. Inside Java By: Siyan, K. S/ Weaver, J. L. New Riders

****

IMT-555(A) MJ: Web Technology [2(T) 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 , velentinBojinov, David Herron, DiogeResende, 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-556(A)MJP: Practical Based on Web Technologies [2Credits]

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-555(B) MJ &556(B)MJP: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 inOption 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

1.2 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]**

4.1Concept 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 ContractFutures-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

6.4Time Value of Options.

Reference Book:

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

Additional Reference:

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

****

IMT-555(C) MJ &556(C)MJP: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, and 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, arclength 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-581 MJP: 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.

SEMESTER-III
Syllabus

IMT 601 MJ: Ordinary Differential Equations [4T]

Course Description: The course "Differential Equations" focuses on the study of different types of linear and nonlinear differential equations of one and two dimensions. It explores the Lorenz model, quadratic ordinary differential equations, and other relevant topics. The course also covers the existence and uniqueness theorem for solutions of differential equations.

Course Objectives:

- To introduce students to the practical significance of solving differential equations.
- To enable students to differentiate between initial value and boundary value problems.
- To emphasize the importance of establishing the existence and uniqueness of solutions
- To equip students with the knowledge and skills to identify and apply appropriate solution methods
- To develop students' ability to classify differential equations based on their characteristics and properties
- To enable students to analytically solve a wide range of ordinary differential equations
- To familiarize students with special functions and their applications
- To ensure students understand the concepts of existence and uniqueness of solutions

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

- Understand the practical importance of solving differential equations, including their applications in modelling real-world phenomena such as population dynamics, growth and decay processes, and physical systems.
- Differentiate between initial value and boundary value problems (IVPs and BVPs), and demonstrate proficiency in solving both types of problems using appropriate techniques.
- Appreciate the importance of establishing the existence and uniqueness of solutions, particularly in the context of initial value problems and systems of differential equations.
- Identify and apply appropriate solution methods for various types of problems, including linear equations of the first and second order, equations with constant coefficients, and equations with variable coefficients.
- Classify differential equations based on their characteristics and properties, including their order, linearity, and whether they are homogeneous or non-homogeneous.
- Analytically solve a wide range of ordinary differential equations (ODEs) using appropriate techniques and methods, including separation of variables, integrating factors, reduction of order, and series solutions for equations with regular singular points.
- Apply knowledge of special functions, such as Legendre and Bessel functions, to solve differential equations with specific types of coefficients or boundary conditions.
- Demonstrate an understanding of the existence and uniqueness of solutions to differential equations, including conditions such as Lipschitz continuity and the method of successive approximations.

Course Contents:

Unit 1: Introduction: Linear equations of the first order **[4 Hours]**

1.1. Differential Equations
1.2. Linear equations of First order

Unit 2: Linear equations with constant coefficients **[8 Hours]**

2.1 The second order homogeneous equation
2.2 Initial value problems
2.3 Linear dependence and independence
2.4 The non-homogeneous equation of order 2
2.5 The homogeneous equation of order n

Unit 3: Linear equations with variable coefficients **[12Hours]**

3.1 Initial value problems
3.2 Solutions of the homogeneous equation
3.3 Wronskian and linear independence
3.4 Reduction of order, non-homogeneous equations
3.5 Homogeneous equations with analytic coefficients
3.6 Legendre equation.

Unit 4: Linear Equations with regular singular points **[12 Hours]**

4.1 Euler equation
4.2 Second order equation with regular singular points
4.3 Exceptional cases
4.4 Bessel's equation.

Unit 5: Existence and uniqueness of solutions to first order equations **[12 Hours]**

5.1 Equations with variable separated, exact equations
5.2 Method of successive approximations
5.3 Lipschitz condition
5.4 Approximation to and uniqueness of solutions.

Unit 6: Existence and uniqueness of solutions to systems and n-th order equations **[12 Hours]**

6.1 Complex n dimensional space, Systems as vector equations
6.2 Existence and uniqueness of solutions to systems
6.3 Uniqueness for linear systems and equations of order n.

Reference Books:

1. E. A. Coddington, An Introduction to Ordinary Differential Equations (Prentice- Hall).
2. G. F. Simmons and S. G. Krantz, Differential Equations (Tata McGraw-Hill).
3. Richard Bronson and Gabriel B. Costa, Differential Equations, Schaum's Outlines

IMT 602 MJ: Computer Network
[2 Credits (T)]

Course Objectives:

- To understand basic concepts of Computer Network.
- To make the student familiarize with the basic taxonomy and terminology of the computer networking area.
- To understand Functions of each layer.
- To understand Network Devices & Various Protocols used in each OSI layer.
- To understand fundamental concepts of Network Security.

Course Outcomes: Identify various types of network topologies and protocols

- Describe the functions of each layer of OSI and TCP/IP Model.
- Differentiate between protocols and devices used in OSI & TCP/IP Model.
- Gain basic knowledge of use of cryptography and Network Security.

Course Content

Unit 1: Introduction to Computer Network **[4 Hours]**

1.1 Introduction to Computer Network
1.2 Applications of Computer Network
1.3 Definition of Protocol and Protocol Standards (De facto and De jure)
1.4 Data Flow and Its Types (Simplex, Half Duplex and Full Duplex)
1.5 Types of Computer Network (LAN, WAN, MAN, PAN)
1.6 Network Topologies (Point to Point, Bus, Mesh, Ring, Star, Tree, Hybrid)

Unit II: Network Models: **[4 Hours]**

2.1 OSI Reference Model
2.2 TCP/IP Protocol Suite
2.3 Comparison between OSI and TCP/IP Model
2.4 Functions of each layer

Unit III: Physical Layer: **[4 Hours]**

3.1 Data and Signals: Analog and Digital signals, Transmission Impairment, Data Rate Limits, Performance.
3.2 Data Transmission Media: Guided Media, Unguided Media and Satellites
3.3 Bandwidth Utilization: Multiplexing and Spreading
3.4 Switching: Circuit switching, Message switching & Packet switching

Unit IV: Data Link Layer: **[3 Hours]**

4.1 Data Link Control: Framing, Flow and Error Control, Error Detection and Correction.
4.2 Data Link Layer Protocols-High-Level Data Link Control (HDLC) & Point-to-Point protocol (PPP)
4.3 Wired LAN and Wireless LAN

Unit V: Network Layer: **[3 Hours]**

5.1 Virtual Circuits and Datagram Subnets, Addresses: Address Space, Notations, Classful addressing, Classless addressing, Subnetting and Network Address Translation (NAT)

5.2 Network Layer Protocols- IP, IPV4 and IPV6

Unit VI: Transport Layer: **[3 Hours]**

6.1 Process-to-Process Communication, Addressing: Port Numbers, Encapsulation and Decapsulation, Multiplexing and Demultiplexing, Flow Control, Error Control, Congestion Control

6.2 Transport Layer protocols-TCP & UDP

Unit VII: Application Layer: **[2 Hours]**

7.1 Application Layer Protocols: HTTP, HTTPS, FTP, SMTP, DNS

Unit VIII: Network Devices & Network Security: **[7 Hours]**

8.1 Repeater, Hub, Switch, Router, Bridge, Modem

8.2 Basics of Network Security

8.3 Cryptography: Definitions, Plain Text, Cipher Text

8.4 Types of Cryptography:

8.5 Symmetric Key Cryptography: Traditional Ciphers, Simple Modern Ciphers.

8.6 Asymmetric Key Cryptography: RSA, Security Services, Digital Signatures.

References:

1. Computer Networks, 5/e, Andrew S. Tanenbaum, David J Wetherall, Pearson Education
2. Computer Networking: A Top-Down Approach, 5/e, James F. Kurose, Keith W. Ross, Pearson Education Data
3. Communications and Networking, Forouzan, McGraw-Hill

Additional Resources

1. https://www.geeksforgeeks.org/computer-network-tutorials/
2. https://www.tutorialspoint.com/computer_fundamentals/computer_networking.htm

IMT 603 MJ: Design and Analysis of Algorithm
[4 Credits (T)]

Course Objectives:

- To design an algorithm and demonstrate performance of algorithms with respect to time and space complexity.
- To become familiar with the fundamentals of algorithm analysis and learn how to use asymptotic notation.
- To learn different algorithm designing strategies and methods.
- To evaluate alternative algorithmic methods' effectiveness critically.
- To gain knowledge of and proficiency in designing algorithms within the context of space and time complexity.

Course Outcome:

- Analyse algorithm running times with asymptotic analysis.
- Compare different data structures with each other. Select the right data structure for the given design scenario.
- Create algorithms by utilizing standard paradigms such as Backtracking, Dynamic Programming, Divide and Conquer, and Greedy.
- Select the finest data structure for a given design state by comparing various data structures.

Course Content

Unit 1: Algorithm Essentials: [10 Hours]

1.1. What is Algorithm?

1.2. Algorithm specification: Pseudo code for expressing algorithms, Recursive Algorithm (Towers of Hanoi)

1.3. Performance analysis- Space Complexity, Time Complexity, Asymptotic Notations (Big O, θ, Ω)

1.4. Problems solving of Asymptotic Analysis

1.5. Estimating running time / number of steps of an algorithm

1.6. Performance analysis of searching algorithm- Linear, Binary search

1.7. Performance analysis of sorting algorithms- insertion sort, bubble sort,

Unit 2: Divide-and-conquer: [6 Hours]

2.1 General method, control abstraction

2.2 Min-Max Problem

2.3 Binary search

2.4 Merge sort, Quick sort

2.5 Strassen's matrix multiplication

Unit 3: Greedy method: **[8 Hours]**

3.1 General method
3.2 Job sequencing with deadlines
3.3 Fractional knapsack problem
3.4 Minimum cost spanning trees- Prim's Algorithm, Kruskal's Algorithm
3.5 Optimal storage on tapes
3.6 Optimal merge patterns
3.7 Huffman coding
3.8 Single source shortest path problem: Dijkstra's Algorithm

Unit 4: Dynamic programming: **[9 Hours]**

4.1 General method-Principle of optimality
4.2 0/1 Knapsack Problem- Merge & Purge
4.3 All pairs shortest path-Floyd Warshall Algorithm
4.4 Longest common subsequence
4.5 String editing
4.6 Traveling sales person problem

Unit 5: Backtracking: **[9 Hours]**

5.1 The general method
5.2 Fixed Tuple vs. Variable Tuple Formulation
5.3 n-queens problem
5.4 Graph Colouring
5.5 Sum of subsets
5.6 Hamiltonian cycles

Unit 6: Basic traversal and search techniques **[7 Hours]**

6.1 Graph Representation definition
6.2 Techniques for graphs traversal: Breadth first search and traversal, Depth first search and traversal
6.3 Topological sorting
6.4 Connected components and spanning trees
6.5 Articulation point and Bridge edge

Unit 7: Branch and Bound Technique: **[7 Hours]**

7.1 Branch and bound terms: Definition of live node, E-node, Dead node, Least cost (LC) search, least cost Branch and Bound (LCBB)
7.2 Control Abstraction for LCBB-search
7.3 0/1 knapsack problem using LCBB method (fixed tuple size)
7.4 Travelling Salesman problem using LCBB method (variable tuple size)

Unit 8: Problem Classification: **[4 Hours]**

8.1 The class of P, NP, NP-hard and NP –Complete
8.2 Relationship among P class, NP class, NP-hard and NP –Complete
8.3 Cook's theorem

Reference Books:

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI Learning Private Limited, 2012.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Pearson Education, Reprint 2006.
3. Harsh Bhasin, Algorithms Design and Analysis, Oxford university press, 2016.
4. S. Sridhar, Design and Analysis of Algorithms, Oxford university press, 2014.
5. www.w3schools.com
6. www.tutorialspoint.com
7. www.javatpoint.com
8. www.geeksforgeeks.com

IMT 604 MJP: Data Analysis Using Python
[4 Credits (P)]

Course Objectives:

- To understand the fundamental concepts of exploratory data analysis using Python.
- To find missing values in data and identify the correlation between different variables.
- To learn how to visualize data in different types of charts.
- To understand and interpret results obtained from graphical analysis.

Course Outcome: Oncompletionofthecourse,studentwillbeableto–

- Understand the fundamentals of exploratory data analysis.
- Implement the data visualization using matplotlib and seaborn library.
- Understanding basics of python for performing data analysis.
- Identify and transform erroneous data using different data Wrangling techniques for Analyzing.
- Import, clean, and explore data to perform preliminary analysis.

Course Content

Assignment 1: Working with Numpy

1. Create a 1D array and a 2D array.
2. Perform basic operations on arrays (addition, subtraction, multiplication).
3. Reshape an array into a different shape.
4. Calculate basic statistical measures like mean, median, and sum.
5. Perform matrix operations.
6. Generate random numbers.
7. Filter elements from an array.
8. Access elements and subarrays using indexing and slicing.

Assignment 2: Working with pandas

1. Create empty dataframes and add records.
2. Create dataframe using numpy array, list, dictionary, list of dictionary, zip() function.
3. Load a dataset(csv/excel) using Pandas.
4. Display basic information about the dataset.
5. Filter rows based on a condition and select specific columns.
6. Group data based on a categorical variable and perform aggregation.
7. Merge two DataFrames based on a common column.

Assignment 3: Data Preprocessing

1. Handle Missing Data
2. Check for Null and Duplicate values
3. Change Data Types of dataframe column.
4. Rename dataframe Columns
5. Perform data transformation.
6. Perform data normalization.
7. Perform data standardization.

Assignment 4: Data Visualization

1. Create a basic line plot.
2. Generate a scatter plot.
3. Create a histogram.
4. Generate a box plot.
5. Create a bar chart.
6. Generate a heatmap.
7. Create a pie chart.
8. Plot multiple lines on the same chart.
9. Create a 3D scatter plot.

Assignment 5: Exploratory data analysis

1. Create box plots to visualize the distribution of numeric variables.
2. Generate scatter plots to explore relationships between numeric variables.
3. Calculate and visualize the correlation matrix.
4. Create pair plots for exploring relationships between multiple variables.
5. Identify and visualize outliers in numeric columns.
6. Convert categorical variables into numeric variables

List of Sample Programs:

1. Write a Program in Python to Read and write different types of Files (csv, json, txt etc).
2. Python program to import libraries for loading & read a dataset. (Use head(), tail(), shape, info() , describe() ,columns)
3. Write a python program to reshaping data- Convert categorical data into numerical value using dataset.

8. Using inbuilt dataset women perform the following

4. Implementation of data cleaning -finding missing data, removing and filling missing data.
5. Write a python program implement data wrangling operations- filtering and removing duplication of data.
6. Python program to Implement data transformation -Combine data frames/datasets using join() ,merge() ,concat() etc.
7. Using the inbuilt mtcar dataset perform the following
    a. Display all the cars having 4 gears
    b. Display all the cars having 3 gears and 2 carburettors.

8. Using inbuilt dataset women perform the following
    a) display all rows of dataset having height greater than 120
    b) display all rows of dataset in ascending order of weight
9. Using the inbuilt air quality dataset perform the following.
    a. Find the temperature of day 30 of month 8
    b. Display the details of all the days if the temperature is greater than 90
10. Using iris inbuilt dataset perform the following
    a. Display details of all flowers of type virginica in ascending order of petal length.
    b. Display details of first five flowers of type setosa having maximum petal length.
11. Write a python program to representation of data using Histogram.
12. Using airquality dataset
    a. Create a scatter plot to show the relationship between ozone and wind values by giving appropriate value to color argument
    b. Create a bar plot to show the ozone level for all the days having temperature greater than 70
13. Using inbuilt mtcars dataset
    a. Create a bar plot that shows the number of cars of each gear type.
    b. Draw a scatter plot showing the relationship between wt and mpg for all the cars having 4 gear
14. Write a python program to representation of data using Pie chart.
15. Write a python program to representation of data using Pair plot/chart.
16. Write a python program for analysis of data through Scatter plot.
17. Write a python program to representation of data using bar plot.
18. Write a python program to implement Univariate analysis.
19. Write a python program to implement bivariate analysis.
20. Write a python program to implement Multivariate analysis.
21. Write a python program to implement correlation matrix and plotting a correlation graph using dataset.
22. Write a python program to implement cross tabulation using crosstab( ) function.
23. Python program to implement data transformation - grouping data using group by.
24. Implementation of measures of central tendency (mean, median and mode) using python.
25. Implementation of measures of dispersion (range, variance) using python.
26. Program to get statistical characteristics of dataset using pandas.
27. Python program to implement Simple regression analysis.

ReferenceBooks:

1. "Hands-On Exploratory Data Analysis with Python", Suresh Kumar Mukhiya,Usman Ahmed. Packt Publication.
2. "Python for Data Analysis", Wes Mckinney, O'REILLY Publication. 2017.

Online Resources:

1) https://www.tableau.com/learn/tutorials/on-demand/getting-started
2) https://www.w3schools.com/python/

IMT- 605(A) MJ: Data Mining

[02 Credits (T)]

Course Outcomes (COs):

CO1: To understand the fundamentals of Data Mining.

CO2: Understand the key algorithms available in data mining.

CO3: Overview of the key fundamental concepts and tools available in Data Warehousing.

Unit 1: Introduction to Data Mining [5 Hours]

1.1 Basic Data, knowledge and information, Data mining
1.2 Data Mining Process, Knowledge Discovery in Databases.(KDD) Process
1.3 Data Mining Tasks
1.4 Data Mining Issues.
1.5 Overview of Applications of Data Mining.

Unit 2: Introduction to Data Warehousing [5 Hours]

2.1Basics of Data Warehouse, Characteristics of Data Warehouse, Architecture of DW-Components of Data Warehouse Architecture, Federated data warehouse architecture
2.2 Basics of, Data Cubes, Data mart, Types of Data Mart
2.3 OLAP and OLTP systems, Difference between OLTP Systems and OLAP
2.4 Dimensional Data Modelling-Star, Snowflake and Fact Constellation schemas.

Unit 3:Data Preprocessing, Data Mining Techniques [5 Hours]

3.1 Data Preprocessing Overview, Steps involved in Data Preprocessing,
3.2 Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization
3.3 Data mining techniques- Association rules, Classification, Regression, Clustering, Neural networks and Perceptrons.

Unit 4: Classification and Clustering **[8 hours]**

4.1 Accuracy Measures- Precision, recall, F-measure, confusion matrix, cross-validation
4.2 Decision tree learning- Construction, Performance, Attribute Selection Measures, Issues: Over-fitting, tree pruning methods,
4.3 Bayesian Classification- Bayes Theorem, Naïve Bayes classifier
4.4 Support Vector Machines classifiers
4.5 k-Nearest Neighbors (KNN) classification
4.6 Basics of clustering; Types of clustering algorithm, Partition (k -means clustering),Hierarchical clustering, Types of Hierarchical clustering

Unit 5: Regression and Association Rule Mining **[7 Hours]**

5.1 Meaning of regression, Types of Regression: linear, non-linear and logistic regression.
5.2 Linear regression: simple, multiple linear regression
5.3 Non-linear regression: polynomial regression
5.4 Data Mining Tasks-Frequent item-sets and Association rule mining: Apriori algorithm,Use of sampling for frequent item-set, FP tree algorithm,

Reference Books:

Data Mining: Concepts and Techniques, Han, Elsevier
ISBN: 9789380931913/
2. 9788131205358
3. 2. Margaret H. Dunham, S. Sridhar, Data Mining – Introductory and Advanced Topics, Pearson Education
4. Tom Mitchell, ―Machine Learning‖, McGraw-Hill, 1997
5. Christopher M. Bishop, ―Pattern Recognition and Machine Learning‖, Springer 2006
6. Raghu Ramkrishnan, Johannes Gehrke, Database Management Systems, Second Edition,
7. McGraw Hill International Ian H. Witten, Eibe Frank Data Mining: Practical Machine Learning Tools and Techniques.

Reference Links:

www.coursera.com
www.edx.org
www.datacamp.com
www.udemy.com
www.brightdata.com
www.w3school.com

IMT- 606(A) MJP: Practical Based on Data Mining

[02 Credits (P)]

Assignments

1. Write a script in R to create a list of cities and perform the following

    1) Give names to the elements in the list.

    2) Add an element at the end of the list.

    3) Remove the last element.

    4) Update the 3rd Element
2. Write a script in R to create two vectors of different lengths and give these vectors as input to array and print that array.
3. Write a script in R to create two vectors and add these two vectors.
4. Write a script in R to create two vectors of different lengths and give these vectors as input to array and print third row of second matrix of the array.
5. Write a script in R to create two vectors of different lengths and give these vectors as input to array and print addition of those matrices.
6. Write a script in R to create a simple bar plot of five subjects marks.
7. Write a script in R to create two vectors of different lengths and give these vectors as input to array and print subtraction of those matrices.
8. Write a R Program to calculate Multiplication Table
9. Write a script in R to create two vectors and add, subtract these two vectors.
10. Write an R Program to calculate Decimal into binary of a given number.
11. Build a classification model and usage of weka to use a Decision Tree algorithm to classify whether data from the “glass-database.arff” file. Perform initial preprocessing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.
12. Build a classification model and usage of weka to use a Decision Tree algorithm to classify weather data from the “weather.arff” file. Perform initial preprocessing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.
13. Build a classification model and usage of WEKA to implement J48 algorithm to classify whether data from the “Diabetes.arff” file. Perform initial pre-processing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data. Find out f-Measure of class containers.
14. Build a classification model and usage of WEKA to implement Naïve Bayes algorithm to classify whether data from the “Iris.arff” file on attribute plat growth and leaves. Perform initial pre-processing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.

15. Build a classification model and usage of weka to use a Decision Tree algorithm to classify whether data from the "labour.arff" file. Perform initial preprocessing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.

IMT- 605(B) MJ: DOTNET Programming

[02 Credits (T)]

Prerequisite: Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism. Familiarity with programming language such as C++ and/or Java.

Course Objectives:

- To understand the DOTNET framework.
- Develop deep understanding of C# language features.
- Build strong concepts of OOP's and implement the same in C#.
- To understand the concept of multi-threading & files.
- To understand and implement the controls & properties of Windows forms.
- To develop database centric applications using ADO.NET.

Course Outcomes:

- Demonstrate the features of Dot Net Framework along with the features of C#.
- Analyse Interfaces for real-time applications.
- Implement of Object-Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.
- Implement the application using File handling.
- Design Windows Application using windows control.
- Design Custom Application Using ADO.NET in C#.

Course Content

Unit 1: Introduction to .Net Framework **[2 Hours]**

1.1 Overview of .NET framework &.Net Architecture
- 1.1.1 The Common Language Runtime (CLR)
- 1.1.2 Microsoft Intermediate Language (MSIL) Code
- 1.1.3 Just in Time Compilers (JITers)
- 1.1.4 The Framework Class Library (FCL)
- 1.1.5 The Common Languages Specification (CLS)
- 1.1.6 The Common Type System (CTS)
- **1.1.7** Garbage Collection (GC)

Unit 2: Introduction to .Net Framework **[4 Hours]**

2.1 Basics of C#. Language (Console Application)

- 2.1.1 Namespace, Variables and Expressions
- 2.1.2 Type Conversion, Boxing and Un-boxing
- 2.1.3 Flow Control
- 2.1.4 Functions
- 2.1.5 Debugging and error handling

2.2Array

- 2.2.1 One-dimensional & two-dimensional array
- 2.3 Exception handling
- 2.3.1System Defined and User Defined

Unit 3: OOPS Concept in C#.NET **[5 Hours]**

3.1 Object Oriented Concept
3.2 Object and Classes
3.3 Class properties: Access modifiers, Implementation of class
3.4 Constructor
3.5 Inheritance
3.6 Polymorphism & Interface
3.7 Abstract Class
3.8 Delegates, Multicasting & Anonymous Methods

Unit 4: Data Structure **[2 Hours]**

4.1 ArrayList
4.2 Collection
4.3 Dictionary
4.4 Hash Table

Unit 5: I/O Stream **[3 Hours]**

5.1 Stream Reader, Stream Writer, File Mode
5.2 Opening & Closing File
5.3 Random Access File

Unit 6: Assembly **[2 Hours]**

6.1Components
6.2.NET Assembly features
6.3 Structure of Assemblies,Calling assemblies, private and shared assemblies

Unit 7: Windows Programming **[6 Hours]**

7.1 Windows Forms -Forms (Menus and Tool Bars, SDI and MDI applications, Building MDI applications.
7.2 Basic Controls - Button, TextBox, Label, RadioButton, CheckBoxDateTimePicker, Timer, PictureBox, ComboBox, ListBox, RichTextBox, MonthCalender
7.3 Container & Dialog Control - GroupBox, Panel, Common Dialog boxes, ProgressBar

Unit 8: Database Connectivity using ADO.NET **[6 Hours]**

8.1 ADO.NET Architecture
8.2 Connection object, Commend Object
8.3 Dataset, DataReader&DataAdapter
8.4 SQL Commands (Insert, Delete, Update, Select)
8.5 Accessing Data with ADO.NET
8.6 Datagridview Data Binding: Insert, Update, Delete records
8.7 Introduction to LINQ

Reference Books:

Programming in C#, E. Balagurusamy,Professional C#, Wrox Publication, C# The Complete Reference", Shildt, TMH, Database Programming with C#, By Carsten Thomsen, Apress.

Additional References:

Free Online Courses on Udemy

Basics of Object Oriented Programming with C# ,

Getting Started with C#

Free Online Video - https**://dotnet.microsoft.com/en-us/learn/csharp**

IMT-606(B) MJP: Practical Based on DOTNet
[2 Credits (P)]

DOTNet Programming Assignments

Assignment 1: **[4 Hours]**

C# Introduction

1. Write a C# program to find the factorial of a given number.
2. Write a C# program to check whether a given number is prime or not.
3. Write a C# program to check whether the given string is a palindrome or not
4. Write a C# program to create an MXN matrix and perform the following operation.
    a. Addition
    b. Multiplication
    c. Transpose
5. Write a C# program to create an MXN matrix and perform the following operation.
    a. Upper Triangular
    b. Lower Triangular
    c. Addition of row elements
    d. Addition of column elements
    e. Addition of diagonal elements

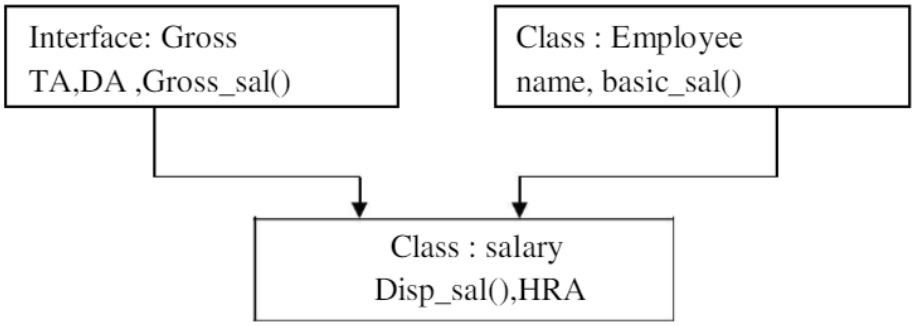
Assignment 2: **[8 Hours]**

OOPs Concepts:

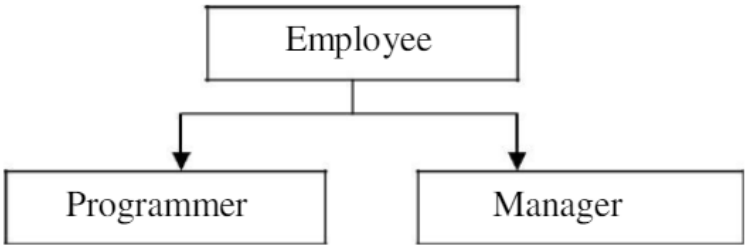
1. Write a program to define a class Students having data members rollno, name. Accept data for 5 student's and display the name of student whose roll no is 3.
2. Implement a base class **Person**. Derive classes **Student** and **Instructor** from **Person**. A Person has aname and a birthday. A student has a batch, course and an Instructor has a salary. Write the class definitions, the constructor and the member function print () for all classes.
3. Write an application that receives the following information from a set of students:

   Student Id:
   Student Name:
   Course Name:
   Date of Birth:
   The application should also display the information of all the students once the data is Entered.

4. Program to implement the following multiple inheritance using interface.



10. Write a program for above class hierarchy for the Employee where the base class is Employee and derived class and Programmer and Manager. Here make display function virtual which is common for all and which will display information of Programmer and Manager interactively.



Assignment 3: **[2 Hours]**

Data Structure

1. Write a C# program to implement a stack with push and pop operations. Find the top element of the stack and check if the stack is empty or not.
2. Write a C# program to find the top and bottom elements of a given stack.

Assignment 4: **[2 Hours]**

I/O Stream

1. C# program to read data from file character by character till the end of the file
2. C# program to compare the content of two files using Stream Reader class
3. C# program to demonstrate the Binary Reader and Binary Writer classes

Assignment 5: **[2 Hours]**

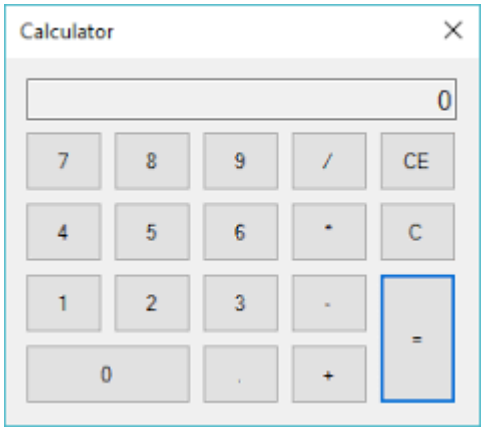
Assembly:

Write a C# program which will demonstrate use of private, public & shared assembly.

Assignment 6: **[6 Hours]**

Windows Programming

Create a windows application to perform following basic arithmetic operations.



1. Create an application that accepts a number from a user in the textbox named num‟. Check whether the number in the textbox num‟ is palindrome or not. Print the message accordingly in the label control named lbldisplay when the user clicks on the button check.

2. Create an application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and checkboxes for selection, the user can make the label text bold, underlined or italic and change its color. include buttons to display the message in the label, clear the text boxes and label and exit.
3. Create a user control that contains a list of colors. Add a button to the Form or testbox which when clicked changes the color of the Form or textbox to the color selected from the list.
4. Create a RadioButtonList that displays the names of some flowers in two columns. Bind a label to the RadioButtonList so that when the user selects an option from the list and clicks on a button, the label displays the flower selected by the user.

Assignment 6: **[6 Hours]**

Database Connectivity using ADO.Net:

1. Write a C# application using ADO.NET to verify if the connection is established with the database or not. Display appropriate messages
2. Write a C# application using ADO.NET to perform insert, delete, update and select operation.

3. Create table Student with the following columns and datatypes.

   Student (rollno Int, NameChar (20), DOB Date)

Insert few records into the table.

Change the candidate name from 'Ram' to 'Krishnan'. Drop the table. Display all the records in gridview.

4. Create table Employee with the following columns and datatypes & perform the following operation
    - **a.** Display all the employees whose SAL is less than 3000.
    - b. Display all the employees who are working as MANAGER or ANALYST.
    - c. Select all the employees who work in department 20 and whose salary exceeds 2000.
    - d. Select the details of employees whose name starts with 'J'.
    - e. Update the salary of employees by 1000 for those drawing less than 2000.
    - f. Find out the average salaries of employees department wise.
5. Create a table "students" with the below given column. Insert records in that & perform the following operation.
    - a. Delete those students who get less than 40 marks.
    - b. Display those students name who get more than 90%
    - c. Display the name of students' whose name starts with ___.

IMT-605(C) MJ: Statistical Inference
[02 Credits (T)]

Course Outcomes: After the completion of this course students will be able:

- Define the terms related to testing of hypotheses.
- Distinguish between parametric and non-parametric tests.
- Explain the procedure of parametric and non-parametric tests.
- Apply an appropriate test to real life situations.
- Use the knowledge of statistical inference to answer the research questions.

UNIT-I: Concepts related to Testing of Hypotheses [4 Hours]

Population, Sample, Parameter, Parameter Space, Statistic, Sampling distribution, Standard error.Concepts of statisticalhypotheses, null and alternative hypothesis, critical region, Type-I error and Type-IIerror, level of significance and power of a test.One and two tailed tests.Differencebetween parametric and non-parametric tests.

UNIT – II: Large sample Tests [6 Hours]

Central Limit Theorem (Statement only), Large sample test forsingle mean and difference of two means, confidence intervals for mean(s). Largesample test for single proportion, difference of proportions.

UNIT – III: Exact Sampling tests [14 Hours]

F-test for equality of variances.t-test for single mean,difference of means and paired t-test, t-test for significance of correlation coefficientand regression coefficient. $\chi 2$-test for goodness of fit and independence of attributes.

UNIT – IV: Non-parametric tests [6 Hours]

One sample run test, sign test and Wilcoxon-signed rank tests(single and paired samples). Two independent sample tests: Median test.

REFERENCE:

1. V.K Kapoor and S.C Gupta, Fundamentals of Mathematical Statistics,
2. E.L. Lehmann and J.P. Romano, Testing Statistical Hypothesis, Fourth Edition
3. P. G. Hoel, Introduction to Mathematical Statistics, Fifth Edition
4. H T Rao: Probability and Statistical Inference. 7th edition.Pearson.

List of Practicals

1. Large Sample Test for Mean
2. Large Sample Test for Proportion
3. F-test for variance
4. t- test for mean
5. t-test for significance of correlation coefficient and Regression coefficients.
6. Chi square test for goodness of fit.
7. Chi square test for independence of attributes.
8. Non-parametric tests.
9. Project.

IMT-606(C) MJP: Statistical Inference
[02 Credits (P)]

1. Large sample test for single mean and difference means
2. Large sample test for single proportion and difference of proportions
3. Large sample test for difference of standard deviations
4. Large sample test for correlation coefficient
5. Small sample test for single mean and difference means
6. Small sample test for correlation coefficient
7. Paired t-test(paired samples).
8. Small sample test for single variance(χ 2 - test)
9. Small sample test for difference of variances(F-test)
10. χ 2 - test for goodness of fit and independence of attributes
11. Non parametric tests for single sample(run test, sign test and Wilcoxon signed ranktest)
12. Non parametric tests for related samples (sign test and Wilcoxon signed ranktest)
13. Non parametric tests for two independent samples (Median test, Wilcoxon –Mann-Whitney - U test, Wald - Wolfowitz' srunstest)

IMT-631: Research Project

[04 Credits]

SEMESTER-IV

IMT-651MJ: Software Engineering and Testing
[Credits 4]

Course Objectives:

- To understand the fundamental concepts of software engineering.
- To learn analysis and design principles for software project development.
- To understand testing methods that can be used as effective tools in quality assurance of software.
- To understand software testing methods and strategies.

Course Outcomes: On completion of the course, student will be able to

- Define and explain the fundamental facts of computer science, software engineering, and multidisciplinary engineering to begin in practice as a software developer.
- Explain the techniques and tools necessary for S/W Development practice.
- Prepare the SRS, Design document, and Project plan of a given software system.
- Identify requirements, analyze, and prepare models.
- Design test cases and test plans to develop quality software.
- Understand where key testing concepts apply within the context of unified processes.

Course Contents

Unit-I: Introduction to Software Engineering and Process Models [10Hours]

1.1 Definition of Software
1.2 Nature of Software Engineering
1.3 Changing nature of software
1.4 Software Process
  1.4.1 The Process Framework
  1.4.2 Umbrella Activities
  1.4.3 Process Adaptation
1.5 Generic Process Model
1.6 Prescriptive Process Models
  1.6.1 The Waterfall Model
  1.6.2 Incremental Process Models
  1.6.3 Evolutionary Process Models
  1.6.4 Concurrent Models
  1.6.5 The Unified Process

Unit-II: Agile Development [12 Hours]

2.1 What is Agility?

2.2 Agile Process
- 2.2.1 Agility Principles
- 2.2.2 The Politics of Agile Development
- 2.2.3 Human Factors

2.3 Extreme Programming (XP)
- 2.3.1XP Values
- 2.3.2XP Process
- 2.3.3 Industrial XP

2.4 Adaptive Software Development (ASD)
2.5 Scrum
2.6 Dynamic System Development Model (DSDM)
2.7 Agile Unified Process (AUP)
2.8 Comparison between Non-Agile and Agile Projects
2.9 Three Stages of Agile Project.
2.10 Tools used in Agile project management: JIRA, Orange Scrum, Kanban, Active Collab, Sprintly, Zepel, Trello, Github Project Management, Pivotal Tracker, Backlog, Wrike, Top Scrum Software Tools in 2022

Unit-III: Requirement Engineering & Design Concepts [6 Hours]

3.1 Requirement Engineering Task:- Inception, Elicitation, Elaboration, Negotiation, Specification, Validation, and Requirements Management.
3.2 Design Concepts: Design Documentation (SRS), Data Design, Architectural Design, Interface Design, Procedural Design.

Unit-IV: UML (Unified Modelling Language) [12 Hours]

4.1 Introduction to UML
4.2 Structural Modelling
- 4.2.1 Use case model
- 4.2.2 Class model

4.3 Behavioural Modelling
- 4.3.1 Sequence model
- 4.3.2 Activity model
- 4.3.3 Communication or Collaboration Model

4.4 Architectural Modelling
- 4.4.1 Component model
- 4.4.2 Artifact model
- 4.4.3 Deployment model

Unit-V: Software Testing Strategies [10 Hours]

5.1 Basics of Software Testing
- 5.1.1 Testing objectives
- 5.1.2 Principles of testing
- 5.1.3 Dichotomies
- 5.1.4 Testing metrics and measurements
- 5.1.5 Verification and Validation
- 5.1.6 Testing Life Cycle

5.2 Testing strategies
- 5.2.1 White Box Testing - Basis path testing, Control Structure Testing.
- 5.2.2 Black Box Testing- Boundary Value Analysis, Equivalence partitioning.
- 5.2.3 Unit testing & Integration testing – Top-down, Bottom-up integration
- 5.3.4 System Testing – Configuration, Compatibility, Performance, Load/Stress testing, Security testing, and Internationalization testing.
- 5.3.5 Acceptance Testing - Alpha, Beta Testing

Unit-VI: Automation Testing & tools [10 Hours]

6.1 Test Management: Test Plan Components, Test Management, Test process, Reporting Test Results

6.2 Defect Management: Finding defects Logging defects tracking and managing Defects, Traceability matrix

6.3 Scope, design and architecture of automation, Introduction to Jira, Selenium etc.

IMT-652MJ: Optimization Techniques
[Credits 4]

Course Objectives:

- To introduce network models and optimization techniques for solving some network related problems.
- To introduce Games and strategies and its applications.
- To learn about sequencing problems and its solution process.
- To introduce the methods to optimize single-variable functions and multi-variable functions.
- To learn about theory of replacement and its utilities.

Course Outcomes:

On completion of the course, student will be able to

- Define and explain the fundamental facts of computer science, software engineering, and multidisciplinary engineering to begin in practice as a software developer.
- Understand and calculate critical path and critical time to complete certain activities.
- Understand practical use of game theory in real life, technique of solving for different types of games.
- Understand sequencing problem with two or three machines and its solutions.
- Learn various optimization methods for optimizing single-variable functions and multi-variable functions

Course Contents

Unit-I: Network Models [15 Hours]

1.1 Scope and Definition of Network Models
1.2 Minimal Spanning Tree Algorithm
1.3 Shortest-Route Problem
1.4 Maximal flow model
1.5 CPM
1.6 PERT a) Expected time
b) Expected variance
c) Critical activity
d) Critical path

Unit-II: Game Theory [10 Hours]

2.1Introduction
2.2 Two-Person Zero-Sum Games
2.3 Pure Strategies (Minimax and Maximin Principles): Games with Saddle Point
2.4 Rules to Determine Saddle Point
2.5 Mixed Strategies: Games without Saddle Point
2.6 The Rules (Principles) of Dominance
2.7 Solution Methods Games without Saddle Point

Unit-III: Replacement and Maintenance Models [10 Hours]

3.1 Introduction
3.2 Types of Failure
a) Gradual Failure
b) Sudden Failure
3.3 Replacement of Items whose Efficiency Deteriorates with Time
3.4 Replacement of Item that completely Fail

Unit-IV: Sequencing Problems [10 Hours]

4.1 Introduction
4.2 Notations, Terminology and Assumptions
4.3 Processing n jobs through two machines
4.4 Processing n jobs through three machines

Unit-V: Classical Optimization Methods [15 Hours]

5.1Introduction
5.2 Unconstrained Optimization
5.3 Optimizing Single-Variable Functions
5.4 Conditions for Local Minimum and Maximum Value
5.5 Optimizing Multivariable Functions
5.6 Constrained Multivariable Optimization with Equality Constraints
a) Direct Substitution Method
b) Lagrange Multipliers Methods
5.7Constrained Multivariable Optimization with Inequality Constraints
a) Kuhn-Tucker Necessary Conditions

IMT-653MJP: Artificial Intelligence and Machine Learning [Credits 4]

Course Objective

- Define Artificial Intelligence (AI) and Machine Learning (ML) Concepts
- Explain the Mathematics required in the industry to use AI and ML
- Introduce the Frameworks, Libraries from Python Programming used in AI and ML
- Introduce Machine Learning Concepts, Processes
- Introduce Neural Networks Concept
- Introduce Computer Vision so that students can work on Face Detection or any other object detection as well as object tracking projects

Course Outcomes

● Students Understand the AI and ML concepts
● They get Hands on experience on handling unstructured as well as structured data
● They are able to use Python Libraries meant for AI & ML and develop useful tools to solve problems
● They are able to track and identify the various objects by developing Machine Learning Models.
● They are able to use API's or libraries available on internet solve AI & ML problems

Course Content

Unit 1: Foundations of AI and ML [15 Hours]

1.1 Introduction to AI and ML
- i. Definition and historical perspective
- ii. Overview of AI and ML applications

1.2 Mathematics for Machine Learning
- i. Linear algebra
- ii. Probability and statistics
- iii. Calculus

1.3 Programming Basics for ML
- i. Introduction to Python
- ii. Loops
- iii. Functions
- iv. Classes& Modules
- v. Libraries for scientific computing (NumPy, Pandas)

1. 4. Reading And Writing Data using Python Libraries (Numpy& Pandas) (Practical)
- i. Read & Write CSV, Microsoft Excel Files

Unit 2: Machine Learning Basics [15 Hours]

1. Supervised Learning
   - a. Linear Regression
   - b. Logistic Regression
   - c. Decision Trees and Random Forests
2. Unsupervised Learning
   - a. Clustering (K-Means, Hierarchical)
   - b. Dimensionality Reduction (PCA)
3. Reinforcement Learning

Unit 3: Advanced Machine Learning - Deep Learning [15 Hours]

1. Neural networks basics
   - a. Representation
   - b. Computing Output
   - c. Vectorizing Examples
   - d. Supervised Learning with Neural networks
2. Activation functions
   - a. Non-Linear Activation function
   - b. Derivatives of Activation functions
3. Shallow Neural Networks
   - a. Forward Propagation
   - b. Backward Propagation

Unit 4: Practical Applications [10 Hours]

1. Natural Language Processing
    a. Text classification
    b. Named Entity Recognition (NER)
2. Computer Vision
    a. Object detection

Unit 5: Capstone Project [10 Hours]

Students work on a hands-on project applying AI and ML techniques to solve a real-world problem. This can involve data acquisition, preprocessing, model training, and evaluation. Possible Set of projects are as follows:

1. Face Detection with Deep Learning
2. Predicting CO2 Emission Footprint Using AI through Machine Learning 3. Movie Recommendations with Movielens Dataset
4. Stock Price Predictions
5. Human Activity Recognition with Smartphones

Problem Sets:

[1] Foundation of AI & ML - use of Pandas, Numpy Python Libraries Data Source:

https://dravate.com/assets/courses/machinelearning/adult.data.csv

(The original source :https://archive.ics.uci.edu/dataset/2/adult) The data has following features
age: continuous;
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked;

fnlwgt: continuous;

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool;

education-num: continuous;

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse,

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty,

Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces;

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried; race: White, Asian-

Pac-Islander, Amer-Indian-Eskimo, Other, Black; sex: Female, Male;

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands;

salary: >50K, <=50K.

Use Panda to answer following questions,

1. How many men and women (sex feature) are represented in this dataset? 2. What is the average age (age feature) of women?
3. What is the percentage of German citizens (native-country feature)? 4. What are the mean and standard deviation of age for those who earn more than 50K per year (salary feature)
5. and those who earn less than 50K per year?
6. Is it true that people who earn more than 50K have at least a high school education?
7. Display age statistics for each race (race feature) and each gender (sex feature). Use groupby() and describe(). Find the maximum age of men of Amer-Indian-Eskimo race.
8. Among whom is the proportion of those who earn a lot (>50K) greater: married or single men (marital-status feature)? Consider married those who have a marital-status starting with Married (Married-civ-spouse, Married-spouse-absent or Married-AF-spouse), the rest are considered bachelors.
9. What is the maximum number of hours a person works per week (hours-per-week feature)? How many people work such a number of hours, and what is the percentage of those who earn a lot (>50K) among them?
10. Count the average time of work (hours-per-week) for those who earn a little and a lot (salary) for each country (native-country). What will these be for Japan?

[2] Supervised Learning:

Data Source:
https://dravate.com/assets/courses/machinelearning/placement-data-class.csv

University Campus recruitment is a strategy for sourcing, engaging and hiring young talent. Our dataset revolves around the placement season of a Business School, Where it has various factors on candidates getting hired such as work experience,exam percentage etc., Finally it contains the status of recruitment and remuneration details.

- Do a exploratory analysis of the Recruitment dataset
- Do an visualization analysis of the Recruitment dataset
- Prediction: To predict whether a student got placed or not using classification models

[3] Supervised Learning

Data Source: https://dravate.com/assets/courses/machinelearning/iris_flower.csvCreate the model that

can classify the different species of the Iris flower[4] Unsupervised Learning

Data Source:

1. https://dravate.com/assets/courses/machinelearning/random.csv2.https://dravate.com/assets/courses/machinelearning/cluster.csv

Kmeans Clustering

In this assignment students are expected to apply k-means clustering to the provided two data sets. The first data set "cluster.csv" contains the data that needs to be clustered. The second data set "random.csv" contains a randomly distributed reference data whose value ranges match those of the "cluster.csv" data.

You need to perform the following experiments.

1. Cluster the cluster.csv data using "simplekmeans" under the cluster tab, with k set to 1, 2, 3, 4, 5 respectively. The cluster mode should be set using a training set. With each clustering run, you can obtain the "within cluster sum of squared error" for that run in the output.

2. For each k value, we run kmeans for five times, each time set the *seed* value to a different value (please use 1, 2, 3, 4, 5). Among the five runs, choose the one that gives the *lowest* "Within cluster sum of squared error" and record the value. (Note that for k=1, no clustering is run, just record the value once will do.) --- This step should give you a set of values we denote WCSE1, WCSE2, WCSE3,WCSE4,WCSE5 for the cluster.csv data

3. Plot the values of WCSEi for i =1, 2, 3, 4, 5. Can you tell the number of clusters from this plot?

4. Repeat steps 1 and 2 for the “random.csv” data and obtain the “within cluster squared error” for k =1, 2, 3, 4, 5 respectively. Let’s denote them by WCSE’1, WCSE’2, WCSE’3,WCSE’4,WCSE’5.

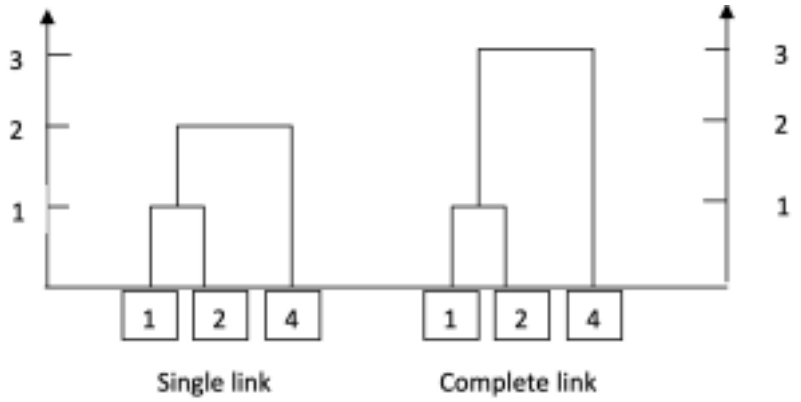
5. Plot the values of WCSEi/WCSE’i, for i =1, 2, 3, 4, 5. From this plot, how many clusters do you think this data contains?

HAC clustering

Create by hand the clustering dendrogram for the following samples of ten points in one dimension.
Sample = (-1.8, -1.7, -0.3, 0.1, 0.2, 0.4, 1.6, 1.7, 1.9, 2.0)

- Using single link
- Using complete link

Note that below are example dendrograms for the following three points (1, 2, 4) with single and complete link.



[5] Reinforcement Machine Learning

A few examples are:

1. Traffic analysis and real-time road processing by video segmentation and frame-by-frame image processing
2. CCTV cameras for traffic and crowd analytics

These examples can involve use of Python Libraries like OpenCV to detect crowds or identify vehicles.

[6] Neural Networks - I

Predict the Burned Area of Forest Fire with Neural Networks.

Data Source: https://dravate.com/assets/courses/machinelearning/forestfires.csvAbout Dataset:

- month : Month of the year: 'jan' to 'dec'
- day : Day of the week: 'mon' to 'sun'
- FFMC : Fine Fuel Moisture Code index from the FWI system: 18.7 to 96.20● DMC : Duff Moisture Code index from the FWI system: 1.1 to 291.3● DC : Drought Code index from the FWI system: 7.9 to 860.6
- ISI : Initial Spread Index from the FWI system: 0.0 to 56.10
- temp : Temperature in Celsius degrees: 2.2 to 33.30
- RH : Relative humidity in percentage: 15.0 to 100
- wind : Wind speed in km/h: 0.40 to 9.40
- rain : Outside rain in mm/m2 : 0.0 to 6.4
- area : The burned area of the forest (in ha): 0.00 to 1090.84

[7] Neural Networks - II

Predicting Turbine Energy Yield (TEY) using Ambient Variables as Features.

The dataset contains 36733 instances of 11 sensor measures aggregated over one hour (by means of average or sum) from a gas turbine.

The Dataset includes gas turbine parameters (such as Turbine Inlet Temperature andCompressor Discharge pressure) in addition to the ambient variables.

Data Source: https://dravate.com/assets/courses/machinelearning/gas_turbines.csv
Reference Books

1. Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, Chen Soon Ong. Book is available at - https://mml-book.github.io/book/mml-book.pdf
2. "Python Machine Learning" by Sebastian Raschka and VahidMirjalili - This book is great for beginners and provides hands-on examples using Python for various ML algorithms.

3. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by AurélienGéron - It covers practical aspects of machine learning using popular Python libraries like Scikit-Learn and TensorFlow.
4. "Pattern Recognition and Machine Learning" by Christopher M. Bishop - This book is a comprehensive introduction to the field, covering both classical and modern techniques.
5. "Deep Learning" by Ian Goodfellow, YoshuaBengio, and Aaron Courville - Often referred to as the "Deep Learning Bible," this book provides a deep dive into deep learning concepts.
6. "Understanding Deep Learning" by Simon J.D.Prince, Published by MIT Press. Available At - https://udlbook.github.io/udlbook/

IMT-654(A) MJ & 655(A) MJP: Go Programming [Credits: 2T+2P=4]

Course Objectives:

- To study various programming constructs in GO
- To understand salient and advance features in GO

Course Outcomes: On completion of the course, student will be able to

- Describe the core features and concepts in Go
- Understand the functions in Go Programming
- Apply defining methods and Go Interfaces
- Use Go routines and Channels
- Explore Go Packages

Unit-I: Introduction [5 Hours]

1.1 Go Runtime and Compilations
1.2 Keywords and Identifiers
1.3 Variables and Constants
1.4 Operators and Expressions and Local Assignments
1.5 Data types in GO
1.6 Pointers and Addresses
1.7 if-else, switches, for loop, Iterations
1.8 breaks and continue
1.9 Strings

Unit-II: Functions [5 Hours]

2.1 Parameters and Return Values
2.2 Call by Value and Reference
2.3 Named Return Variables
2.4 Blank Identifiers
2.5 Variable Argument Parameters
2.6 Using defer statements

2.7 Recursive Functions
2.8 Functions as Parameters

Unit-III: Working with Data [5 Hours]

3.1 Array Literals
3.2 Multidimensional Arrays
3.3 Array Parameters
3.4 Slices and Slice Parameters
3.5 Multidimensional Slices
3.6 Structures and Structure Parameters

Unit-IV: Methods and Interfaces [5 Hours]

4.1 Method Declarations
4.2 Functions vs. Methods
4.3 Pointer and Value Receivers
4.4 Method Values and Expressions
4.5 Interface Types and Values
4.6 Type Assertions and Type Switches
4.7 Method Sets with Interfaces
4.8 Embedded Interfaces
4.9 Empty Interfaces

Unit-V: Go routines and Channels [5 Hours]

5.1 Concurrency vs. Parallelism
5.2 Go routine Functions and Lambdas
5.3 Wait Groups
5.4 Channels
5.5 Sending and Receiving
5.6 Unbuffered and Buffered Channels
5.7 Directional Channels
5.8 Multiplexing with select
5.9 Timers and Tickers

Unit-VI: Packages and Files [5 Hours]

6.1 Packages and Workspaces
6.2 Exporting Package Names
6.3 Import Paths and Named Imports
6.4 Package Initializations
6.5 Blank Imports
6.6 Unit Testing with Test Functions
6.7 Table Tests and Random Tests
6.8 Benchmarking
6.9 Working with Files

IMT-654(B) MJ & 655(B) MJP: DEFI Block Chain Technology

[Credits: 2T+2P=4]

DEFI Block chain Technology [2T+2P]

Subject Code: IMT-654(BMJ) & 655(B) MJP

Course Description:

DEFI Blockchain technology has emerged as one of the most revolutionary innovations of the 21st century, disrupting traditional systems across various industries. This course provides a comprehensive introduction to blockchain technology, covering its principles, applications, and potential impact on society. Throughout this course, students will delve into the fundamental concepts underlying blockchain technology, including decentralized consensus, cryptographic hashing, smart contracts, and distributed ledger technology. They will gain a deep understanding of how blockchain operates, exploring its decentralized nature and its ability to provide transparency, security, and immutability to digital transactions.

Course Objectives:

1. Understand what and why of blockchain technology.
2. Explore how Blockchain works.
3. Learn about Bitcoin, Cryptocurrency and Ethereum.
4. To learn blockchain programming.

Course Outcomes:

On successful completion of this course, students will be able to:
CO1: Learn the fundamentals of Blockchain Technology.
CO2. Learn Blockchain programming
CO3. Basic knowledge of Smart Contracts and how they function.

Course Contents:

Unit 1: Introduction to Blockchain [8]

1.1. Backstory of Blockchain
1.2. What is Blockchain?
1.3. Centralized Vs Decentralized system
1.4. Layers of Blockchain
1.5. Merkle Tree
1.6. Why is Blockchain important?
1.7. Blockchain Uses and Use cases

Unit 2: How Blockchain Works [12]

2.1 Laying the Blockchain Foundation
2.2 Cryptography
2.3 Game Theory
2.4 Properties of Blockchain
2.5 Distributed Consensus Mechanisms
2.6 Blockchain Applications

Unit 3: How Bitcoin Works [5]

3.1 History of money
3.2 Dawn of Bitcoin
3.3 The Bitcoin Blockchain
3.4 The Bitcoin Network
3.5 Full Notes VS SPVs
3.6 Bitcoin Wallets

Unit 4: Smart Contracts [5]

4.1 Ethereum Networks
4.2 Ethereum Virtual Machine, Ether, Gas
4.3 DApps
4.4 Decentralized Autonomous Organizations (DAO)
4.5 Hard and Soft Forks
4.6 Initial Coin Offerings

Textbook

Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions ByBikramadityaSinghal, GautamDhameja, PriyansuSekhar Panda, Apress Media

Reference Books:

- Mastering Blockchain by Imran Bashir, Third Edition, Packt Publication
- Waterhole, The Science of the Blockchain
- Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
- Mastering Ethereum: Building Smart Contracts and DAPPS, by Andreas Antonopoulos, Dr. Gavid Wood, Oreilly Publication

Practical Based on DEFI Blockchain Technology (2 Credit)

Assignment 1: Creating Merkle tree

Assignment 2: Creation of Block

Assignment 3: Block chain Implementation Programming code

Assignment 4: Creating ERC20 token

Assignment 5: Java code to implement blockchain in Merkle Trees

Assignment 6: Java Code to implement Mining using block chain

Assignment 7: Java Code to implement peer-to-peer using block chain

Assignment 8: Creating a Crypto-currency Wallet.

IMT-654(C) MJ and 655(C) MJP: Graph Theory

[Credits: 2T+2P=4]

Course Objectives: To study

- Basics definitions related to Graphs
- Various types of graphs such as Complete, Bipartite, Regular,
- Eulerian, Hamiltonian, etc.
- Trees and connectivity of graphs.
- Shortest path algorithms.

Course Outcomes: On completion of the course, student will be able to

- Convert real life problems into graph model.
- Represent graph as a matrix.
- Find Euler's tour, spanning trees and shortest path in a graph.
- Solve the Chinese postman problem and the travelling salesman problem.
- Understand the concept of directed graphs and their basic results and applications.

Unit-I: An introduction to graphs (8 Hours)

1.1 Definition
1.2 graphs as models
1.3 vertex degrees

1.4 subgraphs
1.5 paths and cycles
1.6 matrix representation of graphs
1.7 fusion.

Unit-II: Trees and Connectivity (8 Hours)

2.1 Definition and simple properties
2.2 bridges
2.3 spanning trees
2.4 connector problems
2.5 shortest path problems
2.6 cut vertices and connectivity.

Unit-III: Euler Tours and Hamiltonian Cycles (8 Hours)

3.1 Euler tours
3.2 Chinese postman problem
3.3 Hamiltonian graphs
3.4 travelling salesman problem.

Unit-IV: Matchings (3 Hours)

4.1Matching and augmenting paths
4.2 marriage problem.

Unit-V Directed Graphs (3 Hours)

5.1 Definitions (and more definitions)

5.2 in degree and out degree

Recommended Book:

1. John Clark and Derrek Allan Holton, A First Look At Graph Theory, Allied Publisher (1995)

Reference Books:

1. Douglas West, Introduction to Graph Theory, Second Edition, Prentice-Hall (2001)
2. Frank Harary : Graph Theory, Narosa Publishers (2001)

IMT-681: Research Project

[Credits: 6]