Scheme for M.Sc. Computer Science NEP 2020

For 2 Year PG programme

First Year

Scheme for M.Sc. Computer Science NEP 2020

For 2 Year PG programme

Scheme sem-1: (For the course of science having major practicum components)

M.Sc. Computer Science I Semester

S.No	Course Code	Course Name	Total Mark	Credit (s)	End Semester Exam Marks		Internal Marks	
					Max	Min.	Max	Min.
1.	MCS2-511	Computer Architecture and Operating System	100	6	60	24	40	16
2.	MCS2-512 A/B	Discrete Structures OR Computer Oriented Optimization Techniques	100	6	60	24	40	16
3.	MCS2-513	Practical – I (Object Oriented Programming using C++)	100	4	60	24	40	16
4.	MCS2-514	Practical – II (Information Security)	100	4	60	24	40	16
5.	MCS2-515	Internship/Seminar/Field Project	100	2	-	-	100	40
		Grand Total	500	22				

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	Computer Architecture and Operating System	MCS2-511

Course learning outcomes (CLO):-

1. Understand basic computer architecture, instruction set and assembly language programming
2. Classify different types, design and architecture of operating system
3. Understand various operating system functions like process management, I/O management, and memory management
4. Solve numerical problems on Process Synchronization, CPU scheduling, Memory management and Disk management.
5. Explore file management, protection and security concepts

Unit-1

A simple computer organization and instruction set, instruction formats, addressing modes, instruction cycle, instruction execution in terms of microinstructions, concepts of interrupt and simple I/O organization, Interrupt Driven data transfer, DMA (Direct Memory Access)., Concept of bus, data movement among registers, conditional data transfer, and data movement from/to memory. Design of simple Arithmetic & Logic Unit & Control Unit, and arithmetic and logical operations. Overview of Architecture of 8086 and assembly language programming.

Unit-2

Introduction: Evolution of operating systems (History of evolution of OS with the generations of computers), Types of operating systems, Multitasking, Timesharing, Multithreading, Multiprogramming and, Real time operating systems, Different views of the operating system, System Programmer's view, User's view, Operating system concepts and structure, Layered Operating Systems, Monolithic Systems. Processes: The Process concept, The process control block, Systems programmer's view of processes, Operating system services for process management, Scheduling algorithms, First come first serve, Round Robin, Shortest run time next, Highest response ratio next, Multilevel Feedback Queues, Performance evaluation of scheduling algorithms stated above

Unit-3

Memory Management: Memory management without swapping or paging, Concepts of swapping and paging, Page replacement algorithms namely, Least recently used, Optimal page replacement, Most recently used, Clock page replacement, First in First out (This includes discussion of Belady's anomaly and the category of Stack algorithms), Modeling paging algorithms, Design issues for paging system, Segmentation, Segmented Paging, Paged Segmentation

Unit-4

Inter-process Communication and Synchronization: The need for inter-process synchronization, Concept of mutual exclusion, binary and counting semaphores, hardware support for mutual exclusion, queuing implementation of semaphores, Classical problems in concurrent programming, Dining Philosopher's problem, Bounded Buffer Problem, Sleeping Barber Problem, Readers and Writers problem, Critical section, critical region and conditional critical region, Monitors and messages. Deadlocks: Concepts of deadlock detection, deadlock prevention, deadlock avoidance. Banker's Algorithm

Unit-5

File System: File systems, directories, file system implementation, security protection mechanisms. Input/output: Principles of I/O Hardware: I/O devices, device controllers, direct memory access. Principles of I/O software: Goals interrupt handlers, device drivers, and device independent I/O software. User space I/O Software. Disks: Disk hardware, Disk scheduling algorithms (namely First come first serve, shortest seek time first, SCAN, C-SCAN, LOOK and C-LOOK algorithms) Error handling, track-at-a-time caching, RAM Disks. Clocks: Clock hardware, memory-mapped terminals, I/O software.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	Discrete Structures	MCS2-512A

Course learning outcomes (CLO):-

1. Understand the basic concepts of set, function and mathematical reasoning.
2. Exposure to various counting techniques and its applications in computer science.
3. Develop an understanding about relations & its property and lattices.
4. Understand the concepts of graph and tree and their applications
5. Develop the understanding of objectives and features of Vedic Arithmetic.

Unit 1

Vedic mathematics :History, salient features, Vedic mathematics formulas, 16 sutras and 13 sub sutras, terms and operations, Beejank, Vinculum Operations, High speed addition by using the concept of completing the whole and superfast subtraction by Nikhilam Sutram. Multiplication by Nikhilam Sutra, Anatyodarshkeyapi, Urdhavtriyaghbhyamsutram,

The Foundations: Logic, Sets and Functions: Introduction to set theory, mathematical logic, prepositions, prepositional equivalences, predicates and quantifiers. Importance of Quantifiers. The Foundations: Logic, Sets and Functions: Sets, set operations, fuzzy sets, functions, functions for computer science, sequences and summations. Logic and computation in ancient Indian texts.

Mathematical reasoning: Introduction to Methods of proof, mathematical induction. Use of mathematical induction to solve different problems. Importance of recursions in computer science, scope of recursions, Recursive definitions, recursive algorithms.

Unit 2

Combinatorics: The basics of counting, The sum rule, The product rule, The Pigeonhole Principle, Permutations with repetitions, Permutations without repetitions, Circular Permutations. Applications of combinations. Applications of Combinatorics to solve Committee problems, word problems, puzzle problems etc. Applications of Combinatorics to understand Telephone numbering plan, understanding Internet addresses, Advanced counting techniques, recurrence relations, solving recurrence relations, algorithm design, Basic understanding of complexities, basic problems of complexity of algorithms.

Unit-3

Relations: Relation definition, Importance of relations in computer science, Relations and their properties, Unary relations, Binary relations, Ternary relations, n-ary relations and their applications, closures of relations, equivalence relations, partial ordering. Representing relations, relation matrix, relation graph, composite relation. Operations on relations – union, intersection and join. Concepts of least upper bond, Greatest lower bond, maximal element, minimal element, Greatest element, Least element of a partially ordered set, lattices, sub lattices, chains

and antichains.

Unit-4

Graphs: Introduction to Graphs, Importance of graph theory in computer science, Graph terminology, representing graphs, graph types, graph models, and graph isomorphism.

Connectivity, Euler and Hamiltonian Paths, shortest path problems, planar graphs, graph colouring, chromatic number, Euler's formula, kuratowski's theorem. The four colour problem, Applications of Graph Colouring, Introduction to Trees, applications of trees, tree traversal, trees and sorting, Spanning trees, minimum spanning trees.

Unit 5

Languages and Grammars: Introduction to Languages and Grammars, solving problems for validity of statements according to the grammar. Importance of Language theory in Computer Science,

Importance of Derivation trees, solving problems of Derivation trees, Importance of Parsing, Phrase-Structure Grammars, Types of Phrase structure grammars.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	Computer Oriented Optimization Techniques	MCS2-512B

Course learning outcomes (CLO):-

1. Get exposure to operations research and linear programming
2. Solve optimization problems
3. Design cost effective and efficient solutions using optimization techniques

Unit 1

Optimization and Indian Knowledge System: Introduction of operation research. LP Formulations, Graphical method for solving LP's with 2 variables, Simplex method, Duality theory in linear programming and applications, Integer linear programming, dual simplex method.

Unit 2

Transportation problem, Assignment problem. Dynamic Programming: Basic Concepts, Bellman's optimality principles, Dynamics programming approach in decision making problems, optimal subdivision problem. Sequencing Models: Sequencing problem, Johnson's Algorithm for processing n jobs through 2 machines, Algorithm for processing n jobs through 3 or more machines, Processing 2 jobs through n machines.

Unit-3

Introduction to Game Theory : Strategy, Minimax and Maximin Criteria, Existence of Saddle Point, Game without Saddle Point, Mixed strategies, Solution of 2x2 games, Rectangular games, Concept and general rules for dominance, Two person zero sum game, Solution of a game by Simplex method.

Unit-4

Integer Programming : Integer Programming, Non – Linear Programming Techniques Kuhn – Tuker conditions with Non – negative constraints, Quadratic programming, Wolfe's Simplex method, Beal's method, Separable Convex programming, Separable Programming Algorithms.

Unit 5

Applications of computer oriented optimization, Optimization in AI and ML, Introduction to the inventory problem, Deterministic Models, The classical EOQ (Economic Order Quantity) model, Inventory models.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – I (Object Oriented Programming using C++).**	MCS2-513

Course learning outcomes (CLO):-

1. Understand and Implement Basic C++ Concepts
2. Use Object-Oriented Features
3. Work with Operators, functions, pointers and Overloading.
4. Design and implement classes and objects.
5. Apply Inheritance, Polymorphism, and Exception Handling
6. Handle File Operations

List of practical

Introduction to Object Oriented Programming
Overview of structured programming approach, Object oriented programming approach, Characteristics of object oriented languages.
Basics of C++ programming :C++ Program Structure, Character Set and Tokens, Data Type, Type Conversion, Preprocessor Directives, Namespace, Input/Output Streams and Manipulators, Dynamic Memory Allocation with new and delete, Control Statements.
Functions: Function Overloading, Inline Functions, Default Argument, Pass by Reference, Return by Reference, Scope and Storage Class.
Pointers: Pointer variables declaration & initialization, Operators in pointers, Pointers and Arrays, Pointer and Function.

Classes & Objects
A Simple Class and Object, Accessing members of class, Initialization of class objects: (Constructor, Destructor), Default Constructor, Parameterized Constructor, Copy Constructor, The Default Copy Constructor, Objects as Function Arguments

Returning Objects from Functions, Structures and Classes, Memory allocation for Objects, Static members, Member functions defined outside the class.

Operator Overloading : Fundamental of operator overloading, Restriction on operator overloading, Operator functions as a class members, Overloading unary and binary operator, Data Conversion (basic to basic, basic to user-defined, user-defined to basic, user-defined to user-defined)
Inheritance : Introduction to inheritance, Derived Class and Base Class, Access Specifiers (private, protected, and public), Types of inheritance, Public and Private Inheritance, Constructor and Destructor in derived classes, Aggregation

Virtual Function, Polymorphism, and miscellaneous C++ Features : Concept of Virtual functions, Late Binding, Abstract class and pure virtual functions, Virtual Destructors, Virtual base class, Friend function and Static function, Assignment and copy initialization, Copy constructor, This pointer, Concrete classes, Polymorphism and its roles.
Function Templates and Exception Handling: Function templates, Function templates with multiple arguments, Class templates, templates and inheritance, Exceptional Handling (Try, throw and catch), Use

of exceptional handling.

File handling : Stream Class Hierarchy for Console Input /Output, Unformatted Input /Output, Formatted Input/Output with ios Member functions, Formatting with Manipulators, Stream Operator Overloading, File Input/output with Streams, Opening and Closing files, Read/Write from File, File Access Pointers and their Manipulators, Sequential and Random Access to File, Testing Errors during File Operations.

USE of IKS in programming

Basic C++ Programs

Swap two numbers, Find the Size of (int, float, double, and char), Fahrenheit to

Celsius, Find Simple Interest, Area And Perimeter of Rectangle

Control Flow Programs

Factorial of a Number, Reverse a Number, Whether a Number is a Palindrome or

Not, Check Armstrong Number, For Fibonacci Number, Make a Simple Calculator

Pattern Printing Programs

Print Simple Full Pyramid Pattern, Inverted Pyramid, Triangle Pattern

Function Programs

Prime Numbers Between Two Intervals Using Function, Check Whether a Number

Can be Express as Sum of Two Prime Numbers, Calculate the Factorial of a Number

Using Recursion

Array Programs

Check if Two Arrays Are Equal or Not, Maximum and Minimum in an Array,

Average of all the Elements Present in an Array, Merge Two ArraysPrint a 2D Array

Matrix Programs

Add Two Matrices, Check Whether Two Matrices Are Equal or Not, Compute the

Sum of Diagonals of a Matrix

Pointers Programs

Array of Pointers, void Pointer, Function Pointer, this Pointer

String Programs

Find the Length of a String, Replace a Character in a String, Compare Two Strings

Class and Object Programs

Create a Class and Object, Encapsulation, Abstraction in Class, Show Data Hiding in Class, Access Modifier, This Keyword in Class, Static keyword

Structures Programs

Pass or Return a Structure to/from a Function, Store Information of a Student in a Structure

Overload Increment ++ and Decrement --, Overload Binary Operator + and -
Add Two Complex Numbers, Show Inheritance

Polymorphism in Class, Function Overloading, Function Overriding, Friend Functions, Virtual Destructor, Abstract Class, Create an Interface

Exception Handling Programs

Show Runtime Exceptions, Handle the Exception Methods, Handle the Checked Exceptions, Handle the Unchecked Exceptions, Handle Divide By Zero and Multiple Exceptions

File Handling Programs

Create a New File, Create a Temporary File, Write Into a File, Rename a File, Make a File Read-Only, Copy one File into Another File, Append the Content of One Text File to Another, Get the List of Files in a Directory, Read Content From One File
and Write it into Another File

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – II (Information Security)**	MCS2-514

Course learning outcomes (CLO):-

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

1. Get hands on experience on Cipher problem related programming
2. Get exposure to various Information and Cyber security related techniques
3. Use cryptography algorithms and protocols to develop secured systems..

List of Practical

Introduction to Information Security : Information Security Mindset, Key characteristics of security mindset, Benefit of security mindset, Attacks, Vulnerability, Security Goals, Security Services and mechanisms.

Conventional Cryptographic Techniques : Conventional substitution and transposition ciphers, One-time Pad, Block cipher and Stream Cipher, Steganography.

Symmetric and Asymmetric Cryptographic Techniques :DES, AES, RSA algorithms Authentication and Digital Signatures : Use of Cryptography for authentication, Secure Hash function, Key management– Kerberos

Program Security : Non malicious Program errors – Buffer overflow, Incomplete mediation, Time-of-check to Time-of- use Errors, Viruses, Trapdoors, Salami attack, Man-in-the- middle attacks, Covert channels
Security in Networks : Threats in networks, Network Security Controls–Architecture, Encryption, Content Integrity, Strong Authentication, Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security–PGP,S/MIME

Caesar Cipher

Affine Cipher with equation = 3x+12

Play fair Cipher with key entered by user.

Poly alphabetic Cipher

AutoKey Cipher

Hill Cipher.

Railfence technique

Simple Columnar Transposition technique

Advanced Columnar Transposition technique.

Simple RSA Algorithm with small numbers.

Simplified DES

Make a study of one IDS(Forex.Snort)

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Internship/Seminar/Field Project**	MCS2-515

Project /Dissertation work (Six weeks) in the form of small thesis based upon the topic related to the M.Sc. Computer Science syllabus . The topics of the dissertation with synopsis will be decided by the

Departmental Research Advisory Committee (DRAC).

M.Sc. Computer Science II Semester

S.No	Course Code	Course Name	Total Mark	Credit(s)	End Semester Exam Marks		Internal Marks	
					Max	Min.	Max	Min.
1.	MCS2-521	Software Engineering	100	6	60	24	40	16
2.	MCS2-522A/B	Internet of Things OR Blockchain Technology	100	6	60	24	40	16
3.	MCS2-523	Practical – I (Data Structures using C++)	100	4	60	24	40	16
4.	MCS2-524	Practical – II (Database Management System)	100	4	60	24	40	16
5.	MCS2-525	Value-added course (VAC) from /MOOCS, SWAYAM, NPTEL	100	2	-	-	100	40
		Grand Total	500	22				

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Software Engineering**	**MCS2-521**

Course learning outcomes (CLO):-

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

1. Understand the applications of software engineering processes and models.
2. Inculcate ability to plan, schedule and estimate software projects.
3. Develop skills for analysis and design of software projects using structured and object oriented approaches.
4. Apply testing and quality assurance mechanisms to produce and reliable system.
5. Ability to develop software systems using software engineering approaches.

Unit-1

Introduction to Software Engineering and Software Processes: Software, Software Classifications and Characteristics, Software Crisis. What is Software Engineering? System Engineering Vs. Software Engineering, Software Engineering Challenges. Software Processes: Process model, Elements and Characteristics of Process model, Process Classification, Software Development Processes: SDLC, Waterfall, Iterative Waterfall, Prototyping, Incremental, Spiral, RAD, Agile Software Development: Principles, Practices & Methods; RUP process, Component-Based Development

model etc..

Unit 2

Project Management and Planning: Project management essentials, Project success and failures, Project Life Cycle, Project team structure and organization, Software Configuration Management, Risk Management. Project planning activities: Metrics and Measurements, Project Size Estimation, Effort Estimation Techniques, Staffing and Personnel Planning, Project Scheduling and Miscellaneous Plans.

Unit-3

Requirements Engineering: Software Requirements, Requirements Engineering Process, Requirements Elicitation. Requirements Analysis: Structured Analysis, Object-oriented Analysis. Requirements Specification, Requirements Validation, and Requirements Management.

Unit-4

Software Design and Coding: Software Design Process, Characteristics of a Good Design, Design Principles, Modular Design (Coupling and Cohesion). Software Architecture. Design Methodologies: Function-oriented Design (Structured Design Methodology in brief). Object oriented Design using UML, Logical Design.

Unit-5

Software Testing, Quality and Maintenance: Testing Fundamentals, Test Planning, Black-Box and White-Box Testing strategy, Levels of Testing, Debugging Approaches. Quality Concept, Quality Factors, Verification and Validation, Quality Assurance Activities, Quality Standards: Capability Maturity Model (CMM). Software Reliability, Software Maintenance and Reengineering.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Internet of Things**	**MCS2-522A**

Course learning outcomes (CLO):-

1. Understand IoT concept
2. Gain knowledge of software components, hardware components and communication technologies involved in IoT.
3. Learn role of cloud computing and security requirements in IoT
4. Develop and evaluate the real life applications of IoT by preparing projects designed with the Arduino.

Unit - I

Introduction to IoT: Definition, Characteristics, IoT design principles, Physical Design of IoT - Hardware and Software components; Logical Design of IoT- functional blocks, IoT communication models, Communication APIs; IoT network architecture, IoT enabling technologies, Introduction to cloud computing in IoT, advantages and disadvantages of IoT, IoT implementation challenges.

Unit – II

Introduction to Arduino Programming: Familiarizing with Arduino Interfacing Board, configuration and architecture, Arduino IDE installation, program structure, data types, variables and constants, operators, control statements and loops, functions, strings, time, arrays, function libraries: I/O functions, Character functions, Math library, Interrupts, Communications. Integration of Sensors and Actuators with Arduino;

Unit – III

Domain specific IoTs: Introduction, home automation, cities, environment, energy, retail, logistics, agriculture, industry, health and lifestyle.

IoT and M2M: Introduction, machine-to-machine Communication, difference between IoT and M2M; SDN and NFV for IoT - Software Defined Networking, Network Function Virtualization.

Unit – IV

Data Acquiring, Organizing and Processing: Introduction, data generation, data acquisition, data validation; Data categorization for storage, various types of data stores, organizing the data, transactions, business processes, integration; Online transactions and processing, stream processing, real-time processing, event stream processing, business process, business intelligence, distributed business process, enterprise systems, service oriented architecture(SOA).

Unit – V

Data Analytics and Machine Learning for IoT: Analytics phases - descriptive, predictive, and prescriptive analytics; Online analytical processing; Introduction to statistical and machine learning tools for data analytics; Introduction to Big data, Big data characteristics, Big data analytics, Apply data analytics to further enhance best practices of Indian Knowledge System

Role of the cloud in IoT: Cloud Storage models and communication APIs for IoT,

Security in IoT: Security challenges for IoT, IoT security practices.

Minor Projects: Sample projects in Arduino: Agriculture, Healthcare, SCM, Connected Cars, Smart city, Smart Home. Application of Data Analysis in Astrology, Ayurveda etc.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Blockchain Technology**	**MCS2-522B**

Course learning outcomes (CLO):-

1. Understand the evolution and purpose of blockchain and distributed ledger technologies.
2. Explain key blockchain components including data structures, hashing, and digital signatures.
3. Understand and compare different consensus algorithms used for block validation.
4. Explore the concepts behind cryptocurrencies, Bitcoin, Ethereum, and smart contracts.
5. Apply the knowledge of blockchain in real-world applications and analyze current trends and challenges.

Unit - I

Foundations of Blockchain and Distributed Systems - Introduction: Digital trust, decentralization, and transparency - History and need for blockchain - Centralized vs. decentralized systems - Distributed ledger technology (DLT) - Blockchain vs. Traditional Databases - Use-case discussion: Land records, healthcare, supply chains

Unit – II

Blockchain Architecture, Hashing & Data Structures - Structure of a block: Header, timestamp, nonce, Merkle root- Blockchain linking and immutability - Cryptographic Hash Functions (SHA-256) - Public/Private Key Cryptography- Digital Signatures and Wallets - Merkle Trees and Proof of Integrity

Unit – III

Consensus Models and Network Validation - Importance of consensus in distributed networks - Proof of Work (PoW): Concept and limitations - Proof of Stake (PoS) and variants- Proof of Activity, Burn, and Elapsed Time (PoET)
- Comparative study of consensus mechanisms - Sybil attack resistance and trust models

Unit – IV

Introduction to Cryptocurrencies: Bitcoin &Ethereum - What is cryptocurrency? - Overview of Bitcoin: Transactions, mining, block rewards - Wallets and key management - Ethereum: Overview and differences from Bitcoin - Smart Contracts and Ethereum Virtual Machine (EVM) – Introduction to wallets: MetaMask, Mist

Unit – V

Blockchain Applications, Challenges & Future Scope- Real-world applications: Voting, banking, logistics, identity- Introduction to platforms: Hyperledger, Corda, IOTA (overview only) - Blockchain as a Service (BaaS) – Cloud integration - Scalability, interoperability, and regulatory challenges - research opportunities in blockchain - Interactive tools (e.g., Blockchain Demo, Ganache)

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – I** (Data Structures using C++)	MCS2-523

Course learning outcomes (CLO):-

1. Implement Stack, Queue, Linked List Structures& tree traversals
2. Identify suitable data structures for software design based on required operations in problem domain
3. Solve real life problems by applying suitable data structures
4. Make use of vast classes and integrate them for problem solutions
5. Apply the knowledge of data structure in designing time & space
6. efficient solutions for real life problems.

List of Practicals

Data Structures Overview: Definitions, types (linear vs. non-linear, Homogeneous vs. Non-Homogeneous, Static vs. Dynamic), and applications of Data Structures.

Abstract Data Types (ADTs): Concepts and implementation. Algorithm Analysis: Time and space complexity, Big O notation.

Arrays: One-dimensional, two-dimensional, and multi-dimensional arrays, memory representation, and operations (insertion, deletion, searching, sorting).

Linked Lists: Singly, doubly, and circular linked lists, operations, and applications. Stacks: Operations (Push, Pop, Traverse), applications (Recursive function-call stack, expression evaluation).

Queues: Operations (Insertion, Deletion, Traversal), Circular Queue, Double ended Queue, applications (process scheduling).

Trees: Definition, Binary trees, Binary Search Trees (BSTs), Tree traversals (inorder, preorder, postorder), and applications of Tree, AVL Tree.

Graphs: Definition, Types of Graphs: Directed, Non-Directed, Connected, Non- Connected, Representation (adjacency matrix, adjacency list), graph traversals (BFS, DFS), and applications, Minimum Spanning Tree, Kruskal's Algorithm, Prim's Algorithm, Dijkastra's Algorithm.

Heaps: Binary heaps, heap operations (insertion, deletion), and applications (priority

queues).

Sorting and Searching Algorithms:Sorting Algorithms: Bubble sort, insertion sort, selection sort, merges sort, quick sort, and their time complexities.

Searching Algorithms: Linear search, binary search, and their time complexities. Hashing: Hash tables, collision resolution techniques.

Specialized Data Structures: B-trees, B+ trees, and their applications. File Structures: Sequential, indexed, and direct access files.

Address calculation of an element in one and two dimensional array (row major order and column major order).

Program for sparse matrix implementation.

Linked Lists :Singly Linked Lists, Circular Single Linked List, Doubly Linked List,

Circular Doubly Linked List

Stack :Stack and its operations PUSH & POP, Stack using Two Queues, Check for Balanced Parentheses, Convert Decimal Number to Binary, Evaluate an Expression, Tower of Hanoi using Binary Value, Program to Solve Tower of Hanoi

Queue :Queue and its operations INSERT & DELETE, Circular Queue, Doubly Ended Queue, Queue using Two Stacks

Tree: Binary Search Tree and its operations, Self Balancing Binary Search Tree Expression Tree from Infix Expression, Find Deepest Left Leaf in a Binary Tree Mirror Image of a Binary Tree,AVL Tree

Graph: Adjacency Matrix, Adjacency List, Inverse of a Graph Matrix, Transpose of a Graph Matrix, Number of Cycles in a Graph, Strongly Connected Components in Graphs, Cycle in a Graph using Graph Traversal, Graph using 2D Arrays, Graph

using Linked List

Searching :Linear, Binary, Interpolation,

Sorting :Bubble,Selection, Insertion,Merge,Quick, .

Hashing : Hash Table, Double Hashing, Chaining with Binary Tree, Linear Probing,Quadratic Probing, Direct Addressing Tables

B-Tree, B+ Tree, Sequential, indexed, and direct access files

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – II** (Database Management System)	MCS2-524

Course learning outcomes (CLO):-

1. Model an application's data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
2. Write SQL commands to create tables and indexes, insert/update/delete data, and query data in a relational DBMS.
3. Learn to optimize SQL queries using query processing and optimization.
4. Understand detailed architecture, define objects, load data, query data, and performance tune SQL databases.
5. Handle large volumes of structured, semi-structured, and unstructured data using database technologies.

List of Practical

Digitization of Archival Records :Background, Image Capturing (Scanning), Image processing and cleaning (Image Enhancement), Optical Character Recognition/ Handwriting Recognition, Translation into English and AI based auto tagging, Conversion to PDF, Subject Metadata and Captioning Unique record identifiers, Quality control checklist, Access to digitized files/records, Long Term Storage, Modalities of digitization, Digitization of Ancient Scriptures and Manuscripts
Introduction: Advantages of DBMS approach, Various views of data, data independence, schema & sub-schema, primary concept of data models, database languages, transaction management, database administrator &user, data dictionary, database architectures. ER model: Basic concept, design issues, mapping constraint, keys, ER diagram, weak & strong entity-sets, specialization &
generalization, aggregation, inheritance, design of ER schema, Reduction of ER Schema to tables. Domains, relation, kind of relation, Relational databases, Various types of keys: candidate, primary, alternate & foreign keys.

Relational Algebra and SQL: The structure, relational algebra with extended operations, modification of database, Idea of relational calculus. Relational Database: Basic structure of SQL, Set operation, Aggregate functions, Null values, Nested Sub queries, derived relations, views, Modification of database, join relation, Domain, relation & keys, DDL in SQL. Programming concepts of PL/SQL, Stored procedure, Database connectivity with ODBC/JDBC

Functional dependencies: Basic definitions, Trivial & non trivial dependencies, closure set of dependencies & of attributes, Irreducible set of dependencies, FD diagram. Normalization: Introduction to normalization, non-loss decomposition, First, second and

third normal forms, dependency preservation, BCNF, multivalue dependencies and fourth normal form, join dependencies and fifth normal form. Database Integrity: general idea, integrity rules, Domain rules, Attributes rules, assertion, triggers, integrity & SQL

Transaction Management: basic concept, ACID properties, transaction state, Implementation of atomicity & durability, Concurrent execution, Basic idea of serializability. Concurrency & Recovery: Basic idea of concurrency control, basic idea of deadlock, Failure Classification, storage structure-types, stable storage implementation, data access, recovery & Atomicity: log based recovery, deferred database modification, immediate database modification, checkpoints.

MongoDB : Introduction, SQL Database, Advantage over RDBMS, Data Types, Data Modeling, MongoDB Operators: Query, Projection, Update Operator Database Commands : Aggregation, Geospatial, Query and Write Operation, Query Plan Cache, Authentication, User Management Role Management, Replication, Shadings, SessionDatabase and Collection : Create, Drop CRUD : Documents (Inset, Update, Delete, Query)
SQL to MongoDB Mapping, MongoDB text search
Digitization of Archival Records :Background, Image Capturing (Scanning), Image processing and cleaning (Image Enhancement), Optical Character Recognition/ Handwriting Recognition, Translation into English and AI based auto tagging, Conversion to PDF, Subject Metadata and Captioning Unique record identifiers, Quality control checklist, Access to digitized files/records, Long Term Storage, Modalities of digitization, Digitization of Ancient Scriptures and Manuscripts
Prepare the case study on ER diagram and normalized database design based on FD's e.g. Retail Banking, Technical Training Institute, Internet Book Shop, Customer Order Warehouse
Design a Database and create required tables. For e.g. Bank, College Database
Apply the constraints like Primary Key , Foreign key, NOT NULL to the tables.
SQL statement for implementing ALTER,UPDATE and DELETE Queries to implement the joins
Query for implementing the following functions: MAX(),IN(),AVG(),COUNT()
Query to implement the concept of Intergrity constrains Query to create the views, Queries for triggers
Perform the following operation for demonstrating the insertion, updation and Deletionusing the referential integrity constraints
Query for creating the users and their role
PL/SQL
Computation, Functions, Procedure, Cursor, Trigger. Where Clause, AND, OR operations in MongoDB.
Commands and Operations of MongoDB in: Insert, Query, Update, Delete and Projection. (Note: use any collection)

M.Sc. Computer Science III Semester

S.No	Course Code	Course Name	Total Mark	Credit (s)	End Semester Exam Marks		Internal Marks	
					Max	Min.	Max	Min.
1.	MCS2-531	Cloud Computing	100	6	60	24	40	16
2.	MCS2-532A/B	Theory of Computation OR Software Project Management	100	6	60	24	40	16
3.	MCS2-533	Practical – I (Web Development using JAVA and Cloud Computing)	100	4	60	24	40	16
4.	MCS2-534	Practical – II(Data Science using Python)	100	4	60	24	40	16
5.	MCS2-535	Internship/Seminar/Field Project	100	2	-	-	100	40
		Grand Total	500	22				

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Cloud Computing**	**MCS2-531**

Course learning outcomes (CLO):-

1. Understand cloud computing concepts and its evolution.
2. Analyze different cloud service models (IaaS, PaaS, SaaS) and deployment models.
3. Gain hands-on experience with AWS/Microsoft Azure/ Google Cloud.
4. Develop understanding of virtualization, cloud storage, security, resource management and edge computing.
5. Plan multi cloud strategies in cloud-based applications

Unit - I

Introduction to Cloud Computing, Evolution from Traditional, Computing to Cloud, Cloud Architecture and Components, Deployment Models: Public, Private, Hybrid, Community Cloud Multi-cloud & Hybrid Cloud Strategies, Advantages & Disadvantages of Cloud Computing.

Unit - II

Cloud Service Models and Virtualization :
Service Models- IaaS, PaaS, SaaS Virtualization-Concepts, Types, Hypervisors, Types of hypervisors Overview of AWS, Microsoft Azure, Google Cloud, IBM Cloud Cloud Resource Management &Scheduling, load balancing, autoscaling
Case Study: Virtual Machine using Virtualbox /Azure, AWS EC2, Google Compute Engine, Microsoft Azure Virtual Machines

Unit – III

Cloud Storage, Security &Compliance :
Cloud Storage: File, Block, Object Storage, Data Management & Providers (AWS S3/ Google Cloud Storage/Azure Blob Storage) Cloud Security Challenges: Data Privacy, Threats, Identity & Access Management (IAM), Compliance & Legal Issues in Cloud, Disaster Recovery & Backup Strategies

Unit – IV

Cloud Container Platforms, Serverless Computing, AI in Cloud Containers: Docker, Kubernetes

Hands-on: AWS Lambda/Google Cloud Functions/Azure Functions Cloud-based AI/ML & Big Data Applications Internet of Things (IoT) and Cloud Computing

Unit – V

Emerging Trends & Future Directions :

Edge Computing & Fog Computing, Blockchain & Quantum Computing in Cloud, Green Cloud Computing and Sustainability, Research Trends & Career Opportunities in Cloud Computing

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Theory of Computation**	**MCS2-532A**

Course learning outcomes (CLO):-

1. Understand computation theory through the use of abstract, formal models.
2. Analyze and compare models of computation, including finite automata, context-free grammars, and turing machines.
3. Learn tools for analyzing the computation models, their strengths and limitations
4. Gain experience with creative mathematical problem solving and develop the ability to write correct, clear, and concise mathematical proofs.

Unit – I

Theory of Automata: String, Alphabet and Languages, Finite Automata, Finite State machine, Basic Definition. Description of a Finite Automaton, Deterministic Finite Accepters Transition Graphs, Languages, Non-deterministic Finite Accepters- Definition, Finite Automata with ϵ- moves, Equivalence of Deterministic and Nondeterministic Finite Accepters, Conversion of NDFA to DFA, Removal of ϵ transition from ϵ – NDFA, Minimization of Finite Automata –Definition and Construction. Mealy and Moore models Definitions, Transformation of Mealy Machine into Moore Machine and vice-versa.

Unit – II

Properties of Regular Sets: Pumping lemma for regular set, Closure properties of regular set. Formal Language: Basic Definition, Chomsky Classification of languages, Regular Expression and Connection between Regular Expressions and Regular Languages.

Unit – III

Regular Grammars – Right and Left Linear Grammars, Equivalence between Regular Languages andRegular Grammars. Context-Free Grammars: Leftmost and Rightmost Derivations, Derivation Trees,Parsing and Ambiguity, Simplification of CFGs. Chomsky Normal Form, Greibach Normal Form, Cocke-Kasami- Younger Algorithm, Properties of Context-Free Languages.

Unit – IV

Pushdown Automata: Definition, Non-deterministic Pushdown Automata, Pushdown Automata for Context Free Languages Context- Free Grammars for Pushdown Automata. Deterministic Pushdown Automata and Deterministic Context-Free Languages.

Unit – V

Turing Machine: Definition of Standard Turing Machine, Turing Machine as Language Accepters and Transducers. Introduction to complexity theory, The Class P and NP.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Theory of Computation**	**MCS2-532B**

Course learning outcomes (CLO):-

1. Learn the skills required for planning, monitoring & evaluation of software projects.
2. Examine Requirements Elicitation, Project Management, Verification &Validation of software projects.
3. Get knowledge to select and apply project management techniques for process modeling, planning, estimation, process metrics and risk management.
4. Gain exposure to tools and techniques of software project management

Unit – I

Introduction to Project Management: What is a Project? What is Project Management? Project phases and project life cycle, Organizational structure, Qualities of a Project Manager, Work Breakdown Structure. Project Management Components: Project Integration Management, Project Plan, Project Development and Execution, Change controls and CCB, Configuration management.

Unit – II

Scope, Time and Cost Management: Strategic planning, Scope planning, definition, Verification and control, Activity planning, Schedule development and control, GANTT Chart, Cost estimation and Control, COCOMO model.

Unit – III

Quality Management and Quality Standards: Quality planning and assurance, CMM levels, KPA‟s, PSP/TSP. **Human Resource Management and Communication Management:** Organizational planning, Staff acquisition, Information distribution, Reporting.

Unit – IV

Risk and Procurement Management: Risk identification, Quantification and control, Solicitation management and control, Contract administration.

Unit – V

Stakeholder Management and Software Metrics: Identifying Stakeholders, Planning, Managing and Monitoring, Stakeholder Engagement, The scope of software metrics, Size-oriented metrics, Function-oriented, Software metrics data collection, Analyzing software data.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – I (Web Development using JAVA and Cloud Computing)**	MCS2-533

Course learning outcomes (CLO):-

1. Develop JAVA programs using the concepts of inheritance, polymorphism, interfaces and packages.
2. Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.
3. Get exposure to advanced java concepts.
4. Develop web based applications using MVC architecture.
5. Implement cloud computing concepts using cloud platforms.
6. Develop applications on cloud platforms
7. Configure cloud security, IAM, and monitoring tools.

List of Practicals

Features of Java, Object-oriented programming overview, Introduction of Java Technologies, How to write simple Java programs, Data Types, Variables, Memory concepts, control statements, looping, Method Call Stack and Activation Record, Argument Promotion and Casting, Scope of declaration and Method Overloading, String Handling: The String constructors, String operators, Character Extraction, String comparison, String Buffer. Arrays: Declaring and Creating Arrays, Enhanced for Statement, Passing Arrays to Method, Multidimensional Arrays, Variable- Length Argument lists, Using Command-line Arguments

Inheritance: Extending classes & related things, Packages and Interfaces: Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, creating own packages Exception Handling: Introduction, overview of doing it and keywords used, when to use it, Multithreading: What are threads, The java Thread model, Thread priorities, Thread life cycle, Thread Synchronization, Applets: Applet basics, Applet Architecture, Applet life cycle methods, Database connectivity: JDBC, The design of JDBC, Typical uses of JDBC
Introduction to HTTP, web Server and application Servers, Installation of Application servers, Config files, Web.xml. Java Servlet, Servlet Development Process, Deployment Descriptors, Generic Servlet, Lifecycle of Servlet, Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlet, Various methods of Session Handling, various elements of deployment descriptors.

JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Connection of JSP and Servlet with different database viz. Oracle, MS- SQL Server, MySQL, java.sql Package, Querying database, adding records, deleting records, modifying records, types

of Statement, Separating Business Logic and Presentation Logic, Building and using JavaBean, Session handling in JSP, Types of errors and exceptions handling.

MVC Architecture Introduction to Remote Method Invocation, Introduction to Enterprise Java Bean, Types of EJB, Creating and working with Session Bean

Class, CLASSPATH, Packages, Exception Handling,

Multithreading, Thread: Life cycle, Synchronization, Applet : Life cycle method, database connectivity, JDBC

Installation of Application servers, Config files, Web.xml. Java Servlet, Deployment Descriptors

Servlet: Generic Servlet, Lifecycle, Packages, Classes, Interfaces, and Methods

Forms with Servlet, Session Handling methods, deployment descriptors.

JSP : Lifecycle, Directives, Scripting elements, Implicit objects,

JSP and Servlet : Connection with Oracle, MS-SQL Server, MySQL, java.sql Package, Session handling

Querying database : Records (Adding, Deleting , Modifying)

Errors and Exceptions handling

JavaBean : Enterprise Java Bean (EJB)

Session bean : Creation and Working MVC : Remote Method Invocation

Cloud Computing

Introduction to Cloud Platforms

-Creating virtual machine using Virtual Box or VMWare

- Setting up free-tier accounts on AWS/ Google Cloud/ Azure
- Exploring cloud dashboards and services

Virtualization and Cloud Storage

- Creating and managing virtual machines (AWS EC2, Google Compute Engine)
- Implementing cloud storage (AWS S3, Google Cloud Storage)

Cloud Development Deployment and Management

- Develop web application through PAAS

- Deploying web applications through Github/ Github Actions
- Serverless computing with AWS Lambda / Google Cloud Functions
- Load balancing and auto-scaling

Security & Monitoring in Cloud

- Identity and Access Management (IAM)
- Cloud security groups, encryption, and compliance tools
- Cloud monitoring and logging (AWS Cloud Watch/ Google Stackdriver)

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical – II** (Data Science using Python)	MCS2-534

Course learning outcomes (CLO):-

1. Understand the fundamentals of Data Science and its applications.
2. Perform data wrangling, data handling, cleaning, and preprocessing techniques using python.
3. Develop machine learning models for prediction and classification tasks.
4. Apply tools & techniques to utilize real-world datasets for decision-making.
5. Apply python based tools & techniques for prediction, analysis, visualisation and classification using real-world datasets

List of Practical

Digital preservation and analysis of ancient text & disease patterns in Ayurveda using Data Science
Introduction to Data Science &Python Basics–, Overview of Data Science and its Applications

1. Python Basics: Data Types, Operators, Control Structures
2. Functions and Modules in Python
3. Introduction to Jupyter Notebook and Google Colab

Data Handling and Preprocessing

1. NumPy for Numerical Computing
2. Pandas for Data Manipulation
3. Data Cleaning: Handling Missing Values, Outliers, and Duplicates
4. Exploratory Data Analysis (EDA)

Data Visualization and Statistical Analysis

- Matplotlib & Seaborn for Data Visualization
- Descriptive and Inferential Statistics
- Correlation and Hypothesis Testing
- Feature Engineering Techniques

Machine Learning using Python

- Introduction to Machine Learning and its Types
- Supervised Learning: Linear & Logistic Regression, Decision Trees

- Unsupervised Learning: Clustering (K-Means, DBSCAN)
- Performance Metrics (Precision, Recall, F1-Score, ROC Curve, Accuracy)

Advanced Topics and Applications

- Introduction to Deep Learning with TensorFlow/Keras
- Introduction to Image Processing and Natural Language Processing (with specific focus on processing local languages too)
- Real-world Case Studies in Data Science
- Ethical Considerations and Future Trends in Data Science

Digital preservation and analysis of ancient text & disease patterns in Ayurveda using Data Science Python Basics for Data Science

- Introduction to Python, Jupyter Notebook, and Google Colab
- Python Libraries: NumPy, Pandas

Data Handling and Cleaning

- Data Preprocessing (Missing Values, Outliers, Duplicates)
- Data Transformation (Encoding, Normalization, Scaling)

Data Visualization and Statistics

- Plotting and visualizing data using Matplotlib and Seaborn
- Performing statistical analysis and correlation tests

Machine Learning Models

- Implementing Linear Regression and Logistic Regression
- Decision Trees and Random Forests
- Unsupervised Learning (K-Means Clustering)

Advanced Applications in Data Science

- Working with real-world datasets (Kaggle)
- Image Processing and Natural Language Processing with Python
- Project in Data Science

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Internship/Seminar/Field Project**	MCS2-535

Project /Dissertation work (Six weeks) in the form of small thesis based upon the topic related to the M.Sc. Computer Science syllabus . The topics of the dissertation with synopsis will be decided by the

Departmental Research Advisory Committee (DRAC).

M.Sc. Computer Science IV Semester

S.No	Course Code	Course Name	Total Mark	Credit (s)	End Semester Exam Marks		Internal Marks	
					Max	Min.	Max	Min.
1.	MCS2-541	Research Methodology	100	6	60	24	40	16
2.	MCS2-542A/B	Compiler Design OR Soft Computing	100	6	60	24	40	16
3.	MCS2-543	Practical – I (Analysis and Design of Algorithms)	100	4	60	24	40	16
4.	MCS2-544	Practical – II (Artificial Intelligence and Machine Learning)	100	4	60	24	40	16
5.	MCS2-545	Value-added course (VAC) from /MOOCS, SWAYAM	100	2	-	-	100	40
		Grand Total	500	22				

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Research Methodology**	MCS2-541

Course learning outcomes (CLO):-

1. Understand the fundamental concepts of research, research questions, hypotheses, and variables, conduct a comprehensive literature review to identify relevant studies, synthesize existing knowledge, and identify research gaps.
2. Identify research problems, formulate research questions, and design appropriate methodologies to address these problems.
3. Identify and select appropriate research designs, such as experimental, observational, survey, qualitative, or mixed-methods, based on the research objectives.
4. Apply appropriate data analysis methods, including statistical techniques or qualitative analysis, to draw meaningful conclusions from research data.
5. Develop a well-structure d research proposal, outlining research questions, methodology, expected outcomes, and a rationale for the study.
6. Communicate research findings effectively through written reports, presentations, and academic papers following the principles of research ethics and integrity.

Unit – I

Research, its meaning, objectives and motivation, Research types, its significance, Research Ethics and Integrity, Plagiarism, its types & tools, Research Methods versus Methodology, Criteria of Good Research, Vedas, Upnishads and Darshana as a means of representing knowledge and presenting in the form of research paper

Unit – II

Research Process, literature review, research strategies and methods, Formulation of research problem, techniques for selecting a research problem

Unit – III

Research Design, features of good design, Different Research Designs/Methods: Pure and Applied Research, Exploratory or Formulative Research, Descriptive Research, Diagnostic Research, Evaluation Studies, Action Research, Experimental Research, Analytical Study or Statistical Method, Historical Research, Surveys, Case Study, Field Studies

Unit – IV

Hypothesis, Nature & Characteristics of Hypothesis, Sampling, Basis & its characteristics, Merits and demerits, Sampling method, choice & tradeoffs, Method of data Collections-Observation, Interview,

Questionnaires and Schedules, Collection of Secondary Data, Statistics in Research, Correlation and Regression, Partial Correlation and Association in Case of Attributes, Quantitative and Qualitative Data Analysis Tools

Unit – V

Interpretation, its techniques, Report Writing, Layout of the Research Report, Types of Reports(Research Proposal/Synopsis, Research Paper, and Thesis), Oral Presentation. Publication ethics: definition, introduction and importance, best practices/standards setting initiatives and guidelines: COPE,WAME, etc., Conflicts of interest, Publicationmisconduct:definition,concept,problemsthatleadtounethical behaviour and vice versa, Violation of publication ethics, authorship and contributor ship, Predatory publishers and journal, Open access publications and initiatives, SHERPA/RoMEO online resource to check publisher copyright & self-archiving policies, Software tool to identify predatory publications developed by SPPU, Journal finder/journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc. E-Resources for research: Google Scholar, Shodh Ganaga, ShodhGangotri

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Compiler Design**	MCS2-542A

Course learning outcomes (CLO):-

1. Understand the design and functioning of Lexical Analyzers.
2. Develop syntax analyzers using top-down parsing and tools like YACC and LEX.
3. Implement efficient bottom-up parsers, including SLR, Canonical LR, and LALR.
4. Conduct semantic analysis and create intermediate representations.
5. Apply optimization techniques and manage effective code Generation

Unit – I

Use of sanskrit in language development, panini grammer & compiler, Compiler Structure: Compilers and Translators, Various Phases of Compiler, Lexical Analysis: The role of Lexical Analyzer, A simple approach to the design of Lexical Analyzer, Implementation of Lexical Analyzer.

Unit – II

LMD (Leftmost Derivation), RMD (Rightmost Derivation), Derivation and Parse Tree, Ambiguity, Capabilities of CFG. Basic Parsing Techniques: Top-Down parsers with backtracking, Non- Recursive Predictive Descent Parser, YACC (Yet Another Compiler Compiler), LEX utility tool.

Unit – III

Bottom–up Parsers, Shift-Reduce Parsing, Operator Precedence Parsers, LR Parsers: SLR (Simple LR), Canonical LR (Left-to-right, LALR (Look- Ahead LR).

Unit – IV

Semantic Analysis, Memory overflow underflow, Type matching, mismatching, global local variable. Intermediate Code Generation: Different Intermediate forms: three address code, Quadruples & Triples. Syntax Directed translation mechanism. Control flow, syntax tree, postfix translation.

Unit – V

Optimization and Code Generation: Local optimization, Loop optimization, Basic blocks and flow graphs, DAG, Data flow analyzer, Symbol Table management, Error handler.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Compiler Design**	MCS2-542A

Course learning outcomes (CLO):-

1. Understand soft computing techniques based on human experience.
2. Develop the skills to design, analyze, and perform experiments on real-life problems using different learning Algorithms
3. Conceptualize fuzzy logic and its applications in real-life problems.
4. Develop mathematical foundation to carry out optimization using genetic algorithm.

Unit – I

Soft Computing: Introduction of Soft Computing, Soft Computing Verses Hard Computing, Various Types of Soft Computing Techniques, Application of Soft Computing, Artificial Intelligence: Introduction, Various types of Production System, Characteristics of production system, breadth first search, depth first search techniques, other search techniques like Hill Climbing, Best first search, A* Algorithm, AO* Algorithms and various types of control strategies, knowledge represent issues, prepositional and predicted logic, monotonic and non monotonic reasoning, forward reasoning, backward reasoning, weak and strong slot and filler structures, NLP

Unit – II

Neural Network: Definition, Advantages, Applications and Introduction to ANN: Introduction to ANN, History of Neural Network, Structure and working of Biological Neural Network, Neural net architecture, Models of neuron-Mc Culloch& Pitts model, Perceptron, Applications of neural networks, Comparison 9of BNN and ANN

Learning Algorithms: Learning and Memory, Learning Algorithms, Numbers of hidden nodes, Error Correction and Gradient Decent Rules, Perceptr on Learning Algorithms, Supervised SPPU

Learning Back propagation, Multilayered Network Architectures, Back propagation Learning Algorithm, Feed forward and feedback neural networks, example and applications.

Unit – III

Associative learning: Introduction, Associative Learning, Hopfield network, Error Performance in Hopfield networks, simulated annealing, Boltzmann machine and Boltzmann learning, State transition diagram and false minima problem, stochastic update, simulated annealing.

Competitive learning Neural network: Components of CL network, Pattern clustering and feature mapping network, ART networks, Features of ART models, character recognition using ART network.

Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self- Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification Convolution Neural Network: Building blocks of CNNs, Architectures, convolution /pooling layers, Padding, Strided convolutions, Convolutions over volumes, SoftMax regression, Deep Learning frameworks, Training and testing on different distributions, Bias and Variance with mismatched data distributions, Transfer learning, multi-task learning, end-to-end deep learning, Application of ANN:

Pattern classification – Recognition of Olympic games symbols, Recognition of printed Characters. Neocognitron – Recognition of handwritten characters. NET Talk: to convert English text to speech.

Unit – IV

Fuzzy Set Overview of Conventional Set, Theory Introduction to Fuzzy Sets, Properties of Fuzzy Sets, Operations on Fuzzy Sets, Crisp Relation, Fuzzy Relation, Tolerance and equivalence relation, Fuzzy Tolerance and equivalence relation, Fuzzy Max-Min and Max-Product Composition, Membership Functions, Fuzzification, Defuzzification to crisp sets, λ-Cuts for fuzzy Relations, Fuzzy (Ruled-Based) system, Graphical technique of inference, Membership value assignment-Intuition, Inference.

Unit – V

Genetic Algorithms

Introduction to Genetic Algorithms History of Genetic Algorithms, What is Genetic Algorithms? Strengths and weaknesses of Genetic Algorithms, Traditional Optimization and Search Techniques, Basic terminologies in Genetic Algorithm, Operators in Genetic Algorithm, Simple Genetic Algorithm, Applications of Genetic Algorithms Optimization Problems, Combinational Optimization, Machine Learning, Image Processing.

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical-I** (Analysis and Design of Algorithms)	MCS2-543

Course learning outcomes (CLO):-

1. Compare and analyze different data structures and sorting algorithms.
2. Understand, Analyse & Implement different algorithm design paradigms such as greedy, dynamic programming, backtracking, branch and bound for effective problem-solving.
3. Understand, Analyse & Implement major graph algorithms.
4. Understand time & space complexity of algorithms.

List of Practicals

Introduction: Algorithmic thinking through Indian Knowledge System, Algorithms, 44nalyzing algorithms, Complexity of algorithms, Growth of functions, Asymptotic Notations, Recurrence Relations and their Solution Methods.

Dynamic Programming: 0/1 Knapsack, Longest Common Subsequence (LCS), Matrix Chain Multiplication. Edit distance, All pair shortest paths – Warshal's and Floyd's algorithms, Optimal binary search tree, Bellman-Ford algorithms.

Graph: Graph Traversal-Breadth First Search, Depth First Search Greedy Methods: Fractional Knapsack, Activity Selection Problem. Minimum Spanning trees – Prim's and Kruskal's algorithms. Single source shortest paths – Dijkstra's algorithms.

Sorting and Order Statistics – Divide-Conquer approach with Quick sort, Merge sort, Comparison of sorting algorithms, sorting in linear time (Counting, Radix, Bucket sort), Heap Sort.

Advanced Data Structures: Red-Black Tree-Properties, Insertion, B-Trees-Creation, Insertion, and Deletion. Introduction to Binomial Heaps – Merge, Union Operation, Fibonacci Heaps – Insertion, Finding Minimum Key, Union.

Backtracking: Graph Coloring, n-Queen Problem, Sum of Subset, Branch and Bound: Travelling Salesman Problem, Introduction to P, NP, NP-complete, NP-Hard.

Implement the following sorting algorithm for a given set of elements and determine the time required to sort the elements. The elements can be read from a file or can be generated using the random number generator.

- Quick sort, Merge sort, Counting, Radix, Bucket sort, Heap Sort

Write programs to implement the following data structures:

- Red-Black Tree, B-Trees

Write programs to print all the nodes reachable from a given starting node in a graph using the following traversal methods:

- Breadth First Search (BFS), Depth First Search (DFS)

Write programs to find the optimal solution for the following problems using the Greedy Method:

- Fractional Knapsack, Activity Selection Problem.
- Minimum Spanning trees – Prim's and Kruskal's algorithms.
- Single source shortest paths - Dijkstra's algorithms

Write programs to find the optimal solution for the following problems using the Dynamic Programming approach:

- 0/1 Knapsack
- Longest Common Subsequence (LCS)
- Matrix Chain Multiplication
- Edit distance
- All pair shortest paths – Warshal's and Floyd's algorithms
- Optimal binary search tree

Write programs to find the optimal solution for the following problems using the Dynamic Programming approach:

- Bellman-Ford algorithms
- Graph Coloring
- n-Queen Problem
- Sum of Subset

Write a program to find the optimal solution for the following problems using the Branch and Bound method:

- Travelling Salesman Problem

BRANCH	Subject	Subject Code
M.Sc. (Computer Science)	**Practical-II** (Artificial Intelligence and Machine Learning)	MCS2-544

Course learning outcomes (CLO):-

1. Understand core AI, ML concepts and problem-solving techniques, including search algorithms and real-world applications.
2. Gain insight into knowledge representation, reasoning, and natural language processing for developing expert systems.
3. Understand the fundamentals of machine learning, supervised & unsupervised learning techniques, and performance evaluation metrics.
4. Develop ability to build and train deep learning models through practical implementation of neural networks, including convolutional neural networks.
5. Apply the concepts of advanced neural network architectures like RNNs and LSTMs, along with reinforcement learning principles and in areas like computer vision and NLP.

List of Practical

Indian Knowledge system and AI, Introduction to Artificial Intelligence (AI): Definition, History, and Goals of AI. Problem-Solving in AI: State space search, Production systems, Search strategies: Uninformed search - Breadth-First Search (BFS), Depth-First Search (DFS). Heuristic Search: Hill climbing, Best-first search, A* and AO* search, Constraint Satisfaction, Means-end Analysis. Applications of AI.

Fundamental Concepts in AI: Knowledge Representation: Propositional logic and predicate logic, Semantic networks and frames. Reasoning - Forward chaining, backward chaining, and resolution and its relation to Indian Knowledge System. Sankrit and AI models. Expert Systems and Rule-Based Systems: Bayes' Theorem, introduction to fuzzy logic, Case study: MYCIN.

Natural Language Processing (NLP): Uses of NLP, Syntactic and Semantic Processing, ATN and RTN.

Introduction to Machine Learning (ML): use of ML, convex optimization, data visualization, hypothesis function and testing, data distributions, data preprocessing, data augmentation, normalizing data sets. Types of ML - supervised, unsupervised, and reinforcement learning. Supervised Learning: Linear regression, Logistic regression, Decision trees. Performance Evaluation Metrics: accuracy, precision, recall, and F1-score, Confusion matrices. Unsupervised Learning: Clustering Algorithms – k-means and hierarchical clustering, metrics to evaluate clusters such as silhouette score.

Introduction to Neural Networks: Structure of neural networks. Activation functions like sigmoid, ReLU, etc., weights and bias, loss function, gradient descent, multilayer network, backpropagation, weight initialization, training, testing, unstable gradient problem, auto encoders, batch normalization, dropout, L1 and L2 regularization, momentum, tuning hyper parameters.

Convolutional neural network, flattening, subsampling, padding, stride, convolution layer, pooling layer, loss layer, dance layer 1x1 convolution, inception network, input channels, transfer learning, one shot learning, dimension reductions, implementation of CNN like tensor flow, keras etc.

Recurrent neural network, Long short-term memory, gated recurrent unit, translation, beam search and width, Bleu score, attention model, Reinforcement Learning, RL- framework, MDP, Bellman equations, Value Iteration and Policy Iteration, , Actor- critic model, Q-learning, SARSA. Application of machine learning in computer vision, speech processing, natural language processing etc.

Indian Knowledge System and Artificial Intelligence Algorithms on BFS and DFS and implement it in C++ or Java.

Algorithm for A* search and trace it with an example.

Setup a python environment for Machine Learning with Anaconda.

Python Basic Programming including Python Data Structures such as List, Tuple, Strings, Dictionary, Lambda Functions, Python Classes and Objects and Python Libraries such as Numpy, Pandas, Matplotlib etc.

Brief Study of Machine Learning Frameworks such as Open CV, Scikit Learn, Keras, Tensorflow etc.

The probability that it is Friday and that a student is absent is 3%. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is the probability that a student is absent given that today is Friday? Apply Baye's rule in

python to get the result.

Extract the data from database using python

Implement linear regression using python

Write a program to demonstrate the working of the decision tree based ID3

algorithm by considering a dataset.

Implement K-Means Clustering using Python. Vary the number of k values as follows and compare the results:

i. 1

ii. 3

iii. 5

iv. 7

v. 11