Week 9

- Write a program to delete a node in a binary search tree?
- Write a program to implement the different operations of an AVL tree
- Write a program to implement the different operations of a threaded binary tree.
- Write a program to implement the different operations of a M-way search tree?

Week 10

- Write a program to implement the different operations of a B- tree?
- Write a program in C++ to implement the different operations of a B+tree.
- Write a program in C++ to implement the graph using different representations.

Week 11

- Write a C++ program to illustrate the traversal of a graph using Breadth FirstSearch.
- Write a C++ program to illustrate the traversal of a graph using Depth FirstSearch.
- Write a program in C++ to find the edges of a spanning tree using Prims Algorithm.
- Write a program in C++ to find the shortest path in a graph using Warshalls Algorithm.

Week 12

- Write a C++ program to in C++ to find the shortest path in a graph using Dijkstra's Algorithm.
- Write a C++ program in C++ to implement Euler Graphs?
- Write a program in C++ to implement Hamilton Graphs?

Week 13

- Write a program in C++ to implement Planner Graphs?
- Write a program to C++ to implement Kruskals Algorithm?

Week 14

- Write a C++ program to implement a simple hash table using linear probing to resolve collisions.
- Write a C++ program to create Max and Min heaps?

COURSE OUTCOMES (CO):

CO1: Students will be able to implement and manipulate linear data structures such as arrays, linked lists, and matrices, including operations like insertion, deletion, and traversal.

CO2: Students will demonstrate proficiency in implementing and applying advanced data structures such as stacks, queues, trees (binary trees, AVL trees), graphs, and various heaps (binomial heaps, leftist heaps) to solve complex problems.

CO3: Students will understand and apply different file organization techniques such as sequential, relative, and indexed sequential file organizations, and multiple key file organizations like inverted files and multi-list organizations.

CO4: Students will develop analytical and problem-solving skills by applying appropriate data structures and algorithms to solve practical problems related to data storage, retrieval, and manipulation in computer science applications.

LEVEL OF CO-PO MAPPING TABLE

Cos	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	-	2	-	-	-	2	1	-	-
2	3	3	3	2	2	2	2	2	2	2	2	-
3	2	2	2	1	-	-	-	2	2	-	-	-
4	3	3	2	-	2	1	-	2	2	2	-	-

To be effective from year-2024

COURSE TITLE: Data Science with Python								
Course Code: MCA24202CR						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 56						**External**	80	40
Total number of Practical Hours: 56						**Internal**	20	10
Lecture (L):	**4**	**Practical(P):**	**2**	**Tutorial (T):**	**0**	**Total Credits**	6	

Course Objectives

- Gain a comprehensive understanding of the fundamental concepts, evolution, and scope of data analytics, including big data and different types of analytics.
- Learn the fundamentals of Python programming, including data types, control flow, and essential packages for data analysis.
- Explore key elements of machine learning, including supervised and unsupervised learning, and apply techniques such as regression and classification.
- Understand and apply various classification methods, including logistic regression, K-NN, and SVM, along with model evaluation techniques.

Course Content	TEACHING HOURS
UNIT 1: Foundation of Data Analytics:	**14 Hrs.**
Introduction to Data Analytics , Evolution, Concept and Scopes Big Data, Metrics and Data classification, Data Reliability & Validity, Problem Solving with Analytics Different phases of Analytics in the business and Data science domain Types of Data Analytics - Descriptive Analytics, Predictive Analytics, Prescriptive Analytics, Applications of Data Analytics Text Analytics and Web Analytics, Skills for Business Analytics Concepts of Data Science, Basic Skills for Data Science	
UNIT 2: Fundamentals of python	**14 Hrs.**
Introduction to Python - Editors & Interactive Development Environments; Custom environment settings for Jupyter,Spyder, PyCharm. Basic data types -numeric, string, float, tuples, list, Python Dictionary, sets and their operations Control flow in python - (if-elif-else), loops (for, while) Inbuilt functions for data conversion, Writing user defined functions in Python Important packages – NumPy, SciPy, Scikit-learn, Pandas, Matplotlib, Seaborn, etc; Installing and loading packages in Python Reading and writing data from/to different formats Python Multi-threaded Programming Plotting in python, functions, list comprehensions, Database connectivity in python, Playing with Date Format.	
UNIT 3: Feature Engineering with Machine Learning	**14 Hrs.**
Introduction, Definitions and types of machine learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning. Reinforcement and Transfer Learning Basics of Regression, Classification, Clustering	

Logistic Regression with one variable and with multiple variables, Application to multi-class classification. The problem of Overfitting, Application of Regularization in Linear and Logistic Regression Regularization and Bias/Variance.	
UNIT 4: Classification and Model Evaluation Techniques	**14 Hrs.**
Classification Using Logistic Regression, Logistic Regression vs. Linear Regression Classification using K-NN, Naive Bayes classifier, Decision Trees Linear Classification using Support Vector Machines Non Linear Classification using Support Vector Machines Cross validation types (train & test, bootstrapping, k-fold validation), Model Performance – Training, Validation and testing; Confusion matrices , Basic evaluation metrics, precision-recall, ROC curves.	

Textbooks:

1. **Jake VanderPlas**, *"Python Data Science Handbook"*, O'Reilly Media, 2016
2. **Joel Grus**, *"Data Science from Scratch"*, O'Reilly Media
3. **Madhusree Ghosh**, *"Data Science and Machine Learning"*, Springer

Reference Books:

Lab Manual

Week 1

- Install Python and set up IDEs like Jupyter Notebook or VS Code
- Write a "Hello, World!" program.
- Write a program to perform basic arithmetic operations: addition, subtraction, multiplication, and division.
- Write a program to print your name and age.

Week 2

- Write a program to create variables of different data types (int, float, complex, string) and print their values.
- Write a program to perform string operations: concatenation, slicing, and repetition.
- Write a program to demonstrate arithmetic, logical, and relational operations.

Week 3

- Write a program to create a list, perform slicing, and append elements to it.
- Write a program to demonstrate the use of tuple data type and its operations.
- Write a program to find the length, maximum and minimum value of a list.

Week 4

- Write a program to demonstrate the use of if, else, and elif statements.
- Write a program to print the first 10 natural numbers using a for loop.
- Write a program to print a pattern using nested loops (e.g., a pyramid).

Week 5

- Write a program to iterate over a string, list, and dictionary using loops.
- Write a program to demonstrate the use of while loops.
- Write a program to manipulate loops using pass, continue, break, and else.

Week 6

- Write a program to to define and call a function that adds two numbers.
- Write a program to demonstrate the use of lambda functions.
- Write a program with a function that takes a list as an argument and returns the sum of all its elements.

Week 7

- Write a program to create and import a custom module.
- Write a program to to use an external library (e.g., math or random).
- Write a program to organize code into a package.

Week 8

- Write a program to define a class and create objects.
- Write a program to demonstrate inheritance.
- Write a program to show polymorphism using method overriding.

Week 9

- Write a program to perform basic array operations with numpy arrays.
- Write a program to create and manipulate DataFrame objects using Pandas.
- Write a program to draw basic plots in Python program using Matplotlib.
- Write a program to perform a basic statistical analysis using SciPy.

Week 10

- Write a program to Count the frequency of occurrence of a word in a body of text is often needed during text processing..
- Write a program to compute weighted averages in Python either defining your own functions or using Numpy.
- Write a python program to calculate the mean, median, mode, variance.

Week 11

- Write a program to create a normal curve using python program.
- Write a python program for correlation with scatter plot
- Write a python program to compute correlation coefficient.

Week 12

- Write a program to demonstrate Regression analysis with residual plots on a given data set.
- Write a program to demonstrate the working of the decision tree-based ID3 algorithm.

Week 13

- Write a program to implement the Naïve Bayesian classifier for a sample training data set.
- Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set.

COURSE OUTCOMES (CO):

CO1: Understand and describe the evolution, concepts, and scope of data analytics.
CO2: Identify and classify different types of data analytics and their applications in various domains.
CO3: Utilize Python programming language for data analysis, including data handling and visualization.
CO4: Implement machine learning techniques such as regression, classification, and clustering for data analysis.
CO5: Evaluate the performance of different machine learning models using appropriate metrics.
CO6: Apply advanced classification techniques and understand their applications in real-world scenarios.

CO	PO											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	1	1	1	2	-	2	1	1	1	1	1
2	1	2	1	1	2	1	2	1	2	2	2	2
3	2	1	3	2	3	2	3	1	2	2	2	2
4	2	1	2	2	2	2	1	1	2	2	2	2
5	1	1	3	3	2	2	2	1	2	2	2	2
6	2	1	3	3	2	2	2	1	2	2	2	2

COURSE TITLE: Operating Systems

Course Code: MCA24203CR									
Total number of Lecture Hours: 56 **Total number of Practical Hours: -**						**Examination Scheme**	**T**	**P**
						External	**80**	-
						Internal	**20**	-
Lecture (L):	**4**	**Practical(P):**	**-**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- Understand Fundamental Concepts of Operating Systems
- Develop Skills in Process Management and Synchronization
- Explore Distributed Operating Systems
- Gain Expertise in Deadlocks Management
- Explore Real Time Operating System
- Acquire Skills in Real-Time Task Scheduling

Course Content	**TEACHING HOURS**
UNIT 1: Introduction and Scheduling	**14 Hrs.**
Operating System Overview, Types of Operating Systems; Basic Operating System: Processes, Scheduling criteria, Scheduling Algorithms. Introduction to Distributed Operating System, Processor allocation and scheduling in distributed systems - System Models, Load balancing and sharing approach, fault tolerance. Introduction to Real Time Operating System, Basic OS Principles and Structures review; Real-Time Systems: Basic Model, Characteristics, Hard vs. Soft. Classification of Real-Time Scheduling Algorithms; Common Approaches; Clock Driven; Priority Driven: Earliest Deadline First, Rate Monotonic, Deadline Monotonic	
UNIT 2: Inter-Process Communication and Synchronization	**14 Hrs.**
Interprocess Communication and Synchronization, Classical problems, Critical section, Semaphores, Monitors. Synchronization in Distributed Systems; Clock Synchronization and related algorithms, Logical Clocks. Mutual Exclusion: Centralized & Distributed (Contention & Token) Algorithms. Election Algorithms: Bully Algorithm, Invitation Algorithm. Client Server model; Remote procedure call and implementation issues. Synchronization in RTOS; Resource Sharing among Real-Time Tasks – Contention and Control; Priority Inversion; Priority Inheritance Protocol; Highest Locker Protocol; Priority Ceiling Protocol	
UNIT 3: Memory Management	**14 Hrs.**
Memory Management: Address Spaces, Virtual Memory. Page Replacement Algorithms, Design and Implementation Issues for Paging Systems, Segmentation. General architecture of Distributed Shared Memory systems; Design and implementation issues of DSM; granularity - Structure of shared memory space, consistency models, replacement strategy, thrashing. Memory Technologies in RTOS; Different Classes of Memory, Memory Access and Layout Issues, Hierarchical Memory Organization [5 Lectures]	
UNIT 4: Deadlocks	**14 Hrs.**

Deadlocks characterization, Methods for handling deadlocks; Deadlock Prevention, Avoidance, Detection, Recovery.
Deadlocks in distributed OS; Deadlock Modeling, Handling Deadlocks in Distributed Systems, Deadlock Avoidance, Deadlock Prevention, Deadlock Detection; Centralized Approach for Deadlock Detection, Fully Distributed Approaches for Deadlock Detection, WFG-Based Distributed Algorithm for Deadlock Detection, Recovery from Deadlock, Issues in Recovery from Deadlock.
Deadlocks in RTOS

Textbooks:

1. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, "Operating System Principles", John Wiley.
2. Pradeep K. Sinha , "Distributed Operating Systems : Concepts and Design", PHI
3. Rajib Mall, Real-Time Systems: Theory and Practice (Second Edition), Pearson Education.

Reference Books:

1. Andrew.S. Tanenbaum, "Modern Operating Systems", PHI. Andrew. S. Tanenbaum,"Distributed Operating System", PHI.
2. Andrew S. Tanenbaum, Modern Operating Systems (Third Edition), Pearson Education.
3. David E. Simon, An Embedded Software Primer, Pearson Education.
4. Laplante, P., Real-Time Systems Design and Analysis (Third Edition), IEEE/Wiley Interscience.
5. Jane W.S. Liu, Real-Time Systems (Sixth Edition), Pearson Education.
6. Raj Kamal, Embedded Systems: Architecture, Programming and Design (Third Edition), Tata McGraw-Hill Education

COURSE OUTCOMES (CO):

CO1: Students will understand the fundamental concepts and functions of an operating system.
CO2: Students will develop skills in process management and CPU scheduling techniques.
CO3: Students will acquire comprehensive knowledge of memory management methods and their practical applications.
CO4: Students will achieve proficiency in the principles and design of distributed systems.
CO5: Students will gain expertise in identifying, preventing, and resolving deadlocks.
CO6: Students will acquire expertise in real-time systems.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	1	1	1	2	1	1	2	2	-	2
2	2	2	2	1	2	1	1	2	1	2	1	2
3	2	1	3	2	1	2	2	1	2	2	1	2
4	2	2	2	3	1	1	2	2	1	2	1	2
5	2	2	3	3	2	2	2	3	1	2	2	2
6	3	3	1	3	2	2	2	2	1	2	1	2

COURSE TITLE: Cryptography and Network Security

Course Code: MCA24204DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- To gain a comprehensive understanding of the OSI Security Architecture and fundamental security concepts.
- To develop proficiency in cryptographic techniques and number theory.
- To master key management and authentication protocols.
- To apply cryptographic methods to network security and intrusion detection.

Course Content	TEACHING HOURS
UNIT 1: Security Fundamentals and Number Theory	**14 Hrs.**
Part 1: Information and Network Security Concepts. The OSI Security Architecture: -Security Attacks: Passive Attacks, Active Attacks, Threats and Vulnerabilities, Malware. OWASP top ten vulnerabilities. -Security Services: CIA, AAA, X.800 -Security Mechanisms: Specific security mechanisms and Pervasive security mechanism. **Part 2:** Introduction to Number Theory: Divisibility and the Division Algorithm, The Euclidean Algorithm, Modular Arithmetic, Prime and relatively prime Numbers, Fermat's and Euler's Theorems, Euler's Totient function, Testing for Primality, The Chinese Remainder Theorem. **Part 3:** Introduction to Cryptology, Classical Encryption Techniques, Substitution Techniques: Monoalphabetic Ciphers and Polyalphabetic Ciphers. Transposition Techniques, One Time Pad	
UNIT 2: Modern Cryptographic Techniques and Algorithms	**14 Hrs.**
Part 1: Block Ciphers: Data Encryption Standard: DES Structure, DES Example, the Strength of DES, Advanced Encryption Standard: AES Structure, AES Transformation Functions, AES Key Expansion, An AES Example. Block Cipher Modes of Operation: Electronic CodeBook, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode. **Part 2:** Random Bit Generation and Stream Ciphers: Principles of Pseudorandom Number Generation, Pseudorandom Number Generators, Pseudorandom Number Generation Using a Block Cipher, Stream Ciphers, RC4, Stream Ciphers Using Feedback Shift Registers. **Part 3:** ASYMMETRIC CIPHERS: Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm, Diffie–Hellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.	
UNIT 3: CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS	**14 Hrs.**

Part 1: Cryptographic Hash Functions SHA-1, SHA-3, Applications of Cryptographic Hash Functions, Two Simple Hash Functions. **Part 2:** Message Authentication Codes, MACs Based on Hash Functions: HMAC, MACs Based on Block Ciphers: DAA and CMAC, Authenticated Encryption: CCM and GCM, Key Wrapping, Pseudorandom Number Generation Using Hash Functions and MACs. **Part 3:** Digital Signatures: RSA Digital signature scheme, ElGamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm. Key management and distribution.	
UNIT 4: NETWORK AND INTERNET SECURITY	**14 Hrs.**
Part 1: Secure Shell /Transport-Level Security: Web Security Considerations, Transport Layer Security, HTTPS, Secure Shell (SSH). **Part 2:** Electronic Mail Security, IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload. **Part 3:** Network Endpoint Security: Firewalls, Intrusion Detection Systems.	

Textbooks

1. William, Stalling, Cryptography and Network Security, 8/E." Prentice Hall. (2023).
2. Forouzan, Behrouz A., and Debdeep Mukhopadhyay. Cryptography and network security (Sie). McGraw-Hill Education, 2011.

Reference Books

1. Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.
2. Introduction to Modern Cryptography (Chapman & Hall/CRC Cryptography and Network Security Series) Jonathan Katz , Yehuda Lindell

COURSE OUTCOMES (CO):

Upon successful completion of this course, learners will be able to:

CO1: Explain the fundamental concepts of cryptography, including symmetric and asymmetric encryption, hashing, digital signatures, and key management.

CO2: Understand the historical development and relevance of cryptographic techniques in modern security protocols.

CO3: Apply various cryptographic algorithms, such as AES, RSA, ECC, and SHA, to secure data and communications.

CO4: Analyze and critically evaluate the strengths and weaknesses of different cryptographic protocols, such as SSL/TLS, IPsec, and PGP.

CO5: Design and implement network intrusion detection systems, integrating cryptographic solutions to protect against various security threats and attacks.

LEVEL OF CO-PO MAPPING TABLE

COs	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	3	-	2	-	-	1	-	-	-	2
2	2	1	2	-	-	3	-	2	-	-	-	2
3	3	2	3	-	3	2	1	2	-	-	-	1
4	2	3	2	3	2	1	-	-	-	-	-	-
5	3	3	3	2	3	-	-	-	2	-	-	-
6	3	3	3	2	2	-	-	1	-	-	-	-

COURSE TITLE: Digital Image Processing

Course Code: MCA24205DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Develop a thorough understanding of the fundamental concepts and theories in image processing, including pixel representation, color spaces, and digital image formation
- Equip students with the technical skills to apply various image processing techniques such as image transformations, filtering, enhancement, and segmentation using appropriate software tools.
- Enhance students' ability to analyze and interpret images by implementing feature extraction and pattern recognition methods, and applying these techniques to solve real-world problems.
- Foster the ability to integrate image processing techniques into broader applications, such as computer vision, medical imaging, and multimedia, through project-based learning and case studies.

Course Content	TEACHING HOURS
UNIT 1: Introduction.	**14 Hrs.**
Introduction Digital Image processing, Origins of DIP, Examples, Fundamental steps in DIP, Components of DIP. Fundamentals Elements of visual perception: brightness, contrast, hue, saturation, Mach-band effect; Light and the electromagnetic spectrum. Image formation and digitization concepts; Image Sensing and acquisition; Image samplingand quantization. Basic relationships between pixels: Neighbours of pixel adjacency connectivity, regions andboundaries, Distance measures.	
UNIT 2: Image Enhancement	**14 Hrs.**
Image enhancement in the spatial domain: Background; Point and arithmetic/ logic operations; Some basic grey level transformations; Histogram processing: Equalization,Matching. Mechanics of spatial filtering: Correlation, Convolution; Smoothing spatial filters: Averaging and Weighted-Averaging Filters, Gaussian Filter; Sharpening spatial filters: First and Second Derivatives, Laplacian, Unsharp Masking and High Boost Filtering. Image enhancement in the frequency domain: Background, Introduction to the Fourier transform and the frequency domain, Smoothing Frequency-Domain filters, Sharpening Frequency Domain filters.	
UNIT 3: Image Restoration and Morphological Processing.	**14 Hrs.**

Model of image degradation/restoration process: Noise models; Restoration by spatial filtering: Mean Filters, Order-Statistics Filters; Restoration by frequency domain filtering: Bandreject Filters, Bandpass Filters. Morphological Processing: Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, connected components, thinning, thickening, skeletons,pruning. Color Image Processing: Color Fundamentals, Color Models: RGB, CMY and CMYK, HIS,Conversion from RGB to HSI and vice versa	
UNIT 4: Edge Detection and Segmentation.	**14 Hrs.**
Edge detection: Basic Formulation: Detecting Points and Lines, Edge Models; Gradient and its Properties; Gradient Operators: Roberts, Prewitt, Sobel; Canny Edge Detector; Thresholding: Basic Global Thresholding, Basic Adaptive Thresholding. [6 Lectures] Region based segmentation: Basic Formulation, Region growing, Region splitting and Merging; Segmentation by morphological watersheds: Basic concepts, Dam construction,Watershed Algorithm.	

Textbooks:

1. Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing, Pearson, SecondEdition, 2004.
2. Anil K. Jain. Fundamentals of Digital Image Processing, Pearson 2002.

Reference Books:

1. Principles of Digital Image Processing by Wilhelm Burger.

COURSE OUTCOMES (CO):

CO1: The students will be able to understand the fundamental principles of image processing, including pixel representation and colour spaces.
CO2: Students will be able to apply image transformation techniques such as scaling, rotation, and translation.
CO3: The students will be able to implement and use various image filtering techniques for noise reduction and edge detection.
CO4: Students will be able to apply image enhancement methods to improve image quality, such as histogram equalization and contrast adjustment.
CO5: Students will be able to perform image segmentation using techniques like thresholding and region-based methods.
CO6: Students will be able to extract and analyse key features from images for pattern recognition and classification tasks.

LEVEL OF CO-PO MAPPING TABLE

COs	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	2	1	1	1	2	-	2	-	1	1	1	1
2	1	2	1	1	2	1	2	-	1	2	2	2
3	2	1	3	2	3	2	3	-	2	1	1	1
4	2	1	2	2	2	2	1	-	1	2	2	2
5	1	1	3	3	2	2	2	-	2	1	1	2
6	2	1	3	3	2	2	2	-	2	2	2	1

COURSE TITLE: Decision Support Systems

Course Code: MCA24206DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- **Understand Decision Support Systems (DSS):** Gain a comprehensive understanding of Decision Support Systems, including their importance in enhancing decision-making processes within organizations.
- **Explore Development Methodologies:** Analyse both traditional and alternative methodologies for DSS development, focusing on their applications, advantages, and limitations. Understand how to manage change effectively during the development and implementation phases.
- **Evaluate DSS Technologies and Tools:** Learn about the various technology levels, development platforms, and tools available for DSS. Develop skills in selecting appropriate tools based on specific needs and technological constraints.
- **Study DSS Components and Models:** Understand the core components and characteristics of DSS. Explore different modelling techniques, including static and dynamic models, and how they handle certainty, uncertainty, and risk. Learn to use influence diagrams and construct mathematical models for decision support.
- **Implement Enterprise DSS:** Examine how DSS supports communication, collaboration, and group decision-making within organizations. Explore the role of enterprise information systems and executive support systems in organizational decision-making and transformation.
- **Facilitate Knowledge Management:** Understand the importance of knowledge management initiatives and approaches. Explore how DSS can aid in organizational learning, knowledge management, and the implementation of effective knowledge management strategies.

Course Content	TEACHING HOURS
UNIT 1: Decision Making	**14 Hrs.**
DSS Development Introduction – Traditional and alternative development methodologies - Change Management – DSS Technology Levels and Tools – Development Platforms – Tool Selection..	
UNIT 2: Modeling and Analysis	**14 Hrs.**
Definition – Characteristics and capabilities of DSS – DSS components - Modeling and issues – Static and dynamic models – Certainty, Uncertainty and Risk – Influence Diagrams – Structure of Mathematical models.	
UNIT 3: DSS Development	**14 Hrs.**
Introduction – Traditional and alternative development methodologies - Change Management – DSS Technology Levels and Tools – Development Platforms – Tool Selection.	
UNIT 4: Enterprise DSS and Knowledge Management	**14 Hrs.**

Communication support – Collaboration support - Group support systems and technologies – GSS meeting process – Creativity and idea generation – Enterprise information systems – Evolution – Characteristics and capabilities of executive support systems – Organizational DSS - Organizational learning and transformation – Knowledge management initiatives – approaches – implementation.

Textbooks

1. Efraim Turban, Jay E Aronson, Ting Peng Liang, Decision Support and Intelligent Systems, Prentice Hall of India, 7th Edition 2005.
2. Efraim Turban, Ramesh Sharda, Dursun Delen, Decision support and Business Intelligence systems, Pearson Education, 9th Edition, 2011.

Reference Books: -

1. Decision Support systems for business Intelligence 2nd edition by Vicki L Sauter Willey
2. Elain Rich and Kevin Knight, Artificial intelligence, TMH, 2006

COURSE OUTCOMES (CO):

CO1: Understand concept of managerial decision systems and outline its various phases.
CO2: Demonstrate DSS components and identify sources of data for business intelligence.
CO3: Categorize the methodologies involved in DSS development.
CO4: Analyze evolution of enterprise DSS and knowledge management initiatives.

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	2	1	1	-	-	2	1	2	1	2
2	3	3	2	2	3	2	1	2	1	2	1	2
3	2	3	3	2	2	-	-	1	1	2	2	2
4	3	3	3	3	3	2	2	2	1	2	2	3

COURSE TITLE: Software Project Management

Course Code: MCA24207DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56						External	80	-
Total number of Practical Hours: -						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- To provide fundamental skills of software Project management emphasizing on issues & hurdles associated with delivering successful projects.
- Apply project management concepts through working in a group as team leader or active team member on an IT project.
- Utilize scheduling terminology, techniques, and tools to create accurate and feasible project timelines.
- Develop and use Bar Charts, Milestone Charts, and Gantt Charts for tracking project progress and communicating schedules.

Course Content	TEACHING HOURS
UNIT 1: Introduction to SPM	14 Hrs.
Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope Document, Project Management Cycle, SPM Objectives SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of Project Plan, Structure of a Software Project Management Plan Software Project Estimation, Estimation Methods, Estimation Models, Decision Process	
UNIT 2: Project Organization and Scheduling Project Elements	14 Hrs.
Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle Ways to Organize Personnel, Project Schedule, Scheduling Objectives, Building the Project Schedule, Scheduling Terminology and Techniques Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts	
UNIT 3: Project Monitoring and Control	14 Hrs.
Dimensions of Project Monitoring & Control, Earned Value Analysis Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI) Software Reviews, Types of Review: Inspections, Deskchecks, Walkthroughs, Code Re views	
UNIT 4: Software Quality Assurance	14 Hrs.

Concept of Software Quality, Software Quality Attributes,
Software Quality Metrics and Indicators, The SEI Capability Maturity Model (CMM)
SQA Activities, Formal SQA Approaches: Proof of Correctness, Statistical Quality Assurance, Product versus process quality management,
Introduction, types of contracts, stages in contract, placement, typical terms of a contract, contract management, acceptance.

Textbooks:-

1. Software Project Management, Bob Hughes and Mike Cotterell, McGraw Hill

Reference Books: -

1. Software Project Management A Unified Framework, Walker Royce, Addison-Wesley
2. A practitioner's Guide to Software Engineering, Roger Pressman, Tata McGraw Hill 2014 8th edition.
3. Basics of Software Project Management, NIIT, Prentice-Hall India, Latest Edition

COURSE OUTCOMES (CO):

CO1: Define the principles of project management for developing software.
CO2: Explain various project management scheduling techniques.
CO3: Apply different techniques of project monitoring, control and review.
CO4: Classify various project management tools and estimate the risks involved in project activities.
CO5: Assess issues related to project quality and staffing.
CO6: Discuss the effect of project management practices in an organization

LEVEL OF CO-PO MAPPING TABLE

Cos	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	2	1	1	-	-	2	1	2	1	2
2	3	3	2	2	3	2	1	2	1	2	1	2
3	2	3	3	2	2	-	-	1	1	2	2	2
4	3	3	3	3	3	2	2	2	1	2	2	3

COURSE TITLE: Machine Learning

Course Code: MCA24208DCE				**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 56 **Total number of Practical Hours: -**				**External**	**80**	**-**
				Internal	**20**	**-**
Lecture (L): 4	**Practical(P): 0**	**Tutorial (T): 0**		**Total Credits**	**4**	

Course Objectives:

- Equip students with a deep understanding of core machine learning techniques, including clustering, classification, dimensionality reduction, and neural networks, with a focus on both theoretical concepts and practical implementations.
- Enable students to apply machine learning algorithms to analyze data, build predictive models, and evaluate their performance using appropriate metrics.
- Teach students advanced dimensionality reduction methods to handle high-dimensional data, enhancing their ability to visualize, interpret, and preprocess data for machine learning tasks.
- Guide students in understanding and implementing artificial neural networks and deep learning techniques to solve complex real-world problems, emphasizing hands-on experience with modern tools and libraries.

Course Content	TEACHING HOURS
	14 Hrs.
UNIT 1: Clustering Techniques **Introduction to Clustering**: Definition, types of clustering (hard vs. soft), applications, and importance. **K-Means and Variants**: K-means algorithm, choosing the number of clusters (elbow method), K-means++, and limitations. **Hierarchical Clustering**: Agglomerative and divisive methods, dendrograms, linkage methods (single, complete, average), and practical applications. **Density-Based Clustering**: DBSCAN, key parameters (epsilon, minPts), and comparison with K-means and hierarchical methods.	
	14 Hrs.
UNIT 2: Classification Techniques **Introduction to Classification**: Overview, types of classification problems, binary vs. multi-class classification. **Bayesian Classifiers**: Naive Bayes, assumptions, advantages, limitations, and Bayesian networks. **Decision Trees and Random Forests**: Concept of decision trees, information gain, Gini index, overfitting, pruning techniques, and introduction to Random Forests. **Support Vector Machines (SVM)**: SVM for linearly separable data, kernel methods for non-linearly separable data, hyperplane and margin concepts. **K-Nearest Neighbors (KNN)**: KNN algorithm, choice of K, distance metrics, and performance optimization.	
UNIT 3: Dimensionality Reduction Techniques	14 Hrs.

Introduction to Dimensionality Reduction: Importance, challenges of high-dimensional data, and the curse of dimensionality. **Principal Component Analysis (PCA):** Eigenvalues, eigenvectors, explained variance, and interpretation of PCA components. **Linear Discriminant Analysis (LDA):** Fisher's criterion, maximizing class separability, and LDA vs. PCA. **Feature Selection Methods:** Filter methods, wrapper methods, and embedded methods.	
UNIT 4: Ensemble Learning Methods	**14 Hrs.**
Ensemble Learning: Definition and motivation for ensemble methods, Types of ensemble methods, Advantages of ensemble learning over single models. **Bagging and Random Forests:** Bootstrap Aggregating (Bagging) concept, Random Forests: construction, feature selection, and out-of-bag error estimation, Comparison of Random Forests with Decision Trees. **Boosting Techniques:** Overview of boosting, AdaBoost: algorithm, weight updates, and practical considerations, Gradient Boosting Machines (GBM): concept, learning rate, and overfitting prevention.	

Textbooks

1. Introduction to Machine Learning by Ethem Alpaydin, MIT Press 4th Edition (2020)
2. Pattern Classification by Duda and Hart. John Wiley publication 2nd Edition.
3. Tom M. Mitchell, "Machine Learning", McGraw-Hill, 2010

Reference Books

1. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer 1st Edition (2006)
2. Machine Learning: A probabilistic Perspective, by Kevin P. Murphy, MIT Press 1st Edition (2012)
3. Introduction to Machine Learning by Ethem Alpaydin, MIT Press, 4th Edition (2020)
4. Pattern Classification by Duda and Hart. John Wiley publication, 2nd Edition (2000)
5. The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2nd Edition (2009)

COURSE OUTCOMES (CO):

CO1: Students will demonstrate the ability to implement and evaluate various clustering and classification algorithms, including K-means, hierarchical clustering, decision trees, SVMs, and KNN, applying them effectively to real-world datasets.

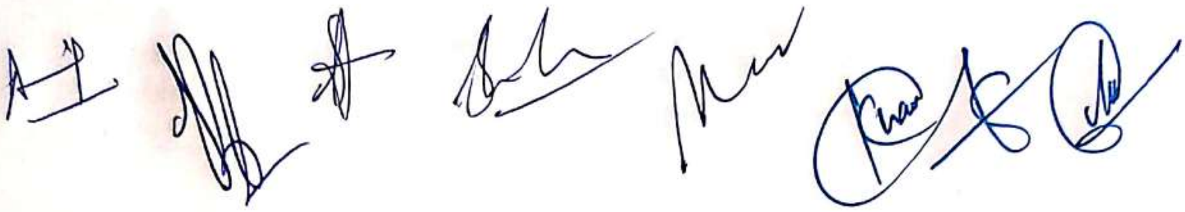
CO2: Students will acquire the skills to apply dimensionality reduction techniques like PCA, LDA, and t-SNE, optimizing models for better performance and interpretability in high-dimensional spaces.

CO3: Students will be able to design, train, and optimize artificial neural networks, including deep learning architectures such as CNNs and RNNs, for applications in areas like image and speech recognition.

CO4: Students will develop the expertise to analyze complex data-driven problems, design appropriate machine learning solutions, and critically evaluate their models using rigorous validation techniques and metrics.

LEVEL OF CO-PO MAPPING TABLE

COs	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	2	3	1	-	-	2	1	-	2
2	3	3	2	2	3	1	-	-	1	1	-	2
3	3	3	3	3	3	2	-	1	2	1	1	3
4	3	3	3	3	3	1	1	1	2	2	2	3



COURSE TITLE: Cloud Computing

Course Code: MCA24209DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Understand core cloud computing concepts and service models.
- Gain practical skills in deploying and managing cloud applications.
- Understand how to manage cloud service performance, reliability, and security.
- Analyze the cost and benefits of different cloud platforms.

Course Content	TEACHING HOURS
	14 Hrs.
UNIT 1: CLOUD COMPUTING FUNDAMENTALS	
Cloud Computing definition;, private, public and hybrid cloud. Cloud types; IaaS, PaaS, SaaS. Benefits and challenges of cloud computing, public vs private clouds, Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery; next generation Cloud Applications.	14 Hrs.
UNIT 2: VIRTUALIZATION AND CLOUD APPLICATIONS	
VIRTUALIZATION: Role of virtualization in enabling the cloud : Types of Virtual Machines, Advantages of Virtualization, Components of Virtualization, CLOUD APPLICATIONS : Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages	14 Hrs.
UNIT 3: MANAGEMENT OF CLOUD SERVICES	
Reliability, availability and security of services deployed from the cloud. Performance and scalability of services, tools and technologies used to manage cloud services deployment; Cloud Economics: Cloud Computing infrastructures available for implementing cloud based services. Economics of choosing a Cloud platform for an organization, based on application requirements, economic constraints and business needs (e.g Amazon, Microsoft and Google, Salesforce.com, Ubuntu and Redhat)	14 Hrs.
UNIT 4: APPLICATION DEVELOPMENT	
Application Development: Design and implementation in cloud environments. Development Platforms: AWS, Azure, Google App Engine. Deployment and management strategies for cloud applications.	

Textbooks

- Gautam Shroff, "Enterprise Cloud Computing: Technology, Architecture, Applications", Cambridge University Press; 2nd Edition [ISBN: 9780521137355], 2023.
- Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing: A Practical Approach" McGraw-Hill Education; 2nd Edition [ISBN: 9780071826400], 2018.
- Dimitris N. Chorafas, "Cloud Computing Strategies" CRC Press; 2nd Edition [ISBN: 9780367338611], 2021.

Reference Books

- Thomas Erl, "Cloud Computing: Concepts, Technology & Architecture" Prentice Hall; 3rd Edition [ISBN: 9780133994164], 2024.
- Rajkumar Buyya, Christian Vecchiola, and Selvi, S. Thamarai, "Mastering Cloud Computing: Foundations and Applications Programming" Morgan Kaufmann; 3rd Edition [ISBN: 9780128180747], 2022.

COURSE OUTCOMES (CO):

CO1: Explain cloud computing principles and service models.
CO2: Successfully deploy and manage cloud-based applications.
CO3: Apply best practices for cloud service management.
CO4: Assess the economic aspects of cloud computing platforms

LEVEL OF CO-PO MAPPING TABLE

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	1	2	3	1	2	-	1	2	-	3
2	2	3	2	3	3	2	1	1	2	2	1	2
3	1	2	3	2	2	3	3	1	3	1	2	3
4	2	1	2	3	1	2	1	3	1	3	3	2

COURSE TITLE: Linux Programming

Course Code: MCA24210DCE						Examination Scheme	T	P
Total number of Lecture Hours: 56 Total number of Practical Hours: -						External	80	-
						Internal	20	-
Lecture (L):	4	Practical(P):	0	Tutorial (T):	0	Total Credits	4	

Course Objectives

- Describe the structure, features and utilities available in Linux
- Use Linux utilities for system administration
- Develop basic applications using Shell scripting
- Describe various methods of extending a Linux kernel
- Develop kernel modules for extending Linux kernel
- Develop GUI applications using Qt programming

Course Content	TEACHING HOURS
UNIT 1: Introduction to Linux	14 Hrs.
Introduction – History, acquisition and installation, Linux features and directory structure. Linux utilities – directory and file manipulation, text processing, process management, system information, creating and managing users, setting ownerships/permissions, managing services.	
UNIT 2: Shell scripting	14 Hrs.
Shell – definition & types. Variables – local, shell & environment. Operators – test, expr, bc, built-in. Floating-point arithmetic. Expressions – arithmetic, relational and logical. Looping & decision-making statements. Substitution – filename, variable and command. Functions and positional parameters. Writing shell scripts for developing basic applications.	
UNIT 3: Kernel development	14 Hrs.
Linux kernel architecture. Building the kernel. Extending the kernel -- Syscalls and kernel modules. Compiling Modules. Loading/unloading modules. Module licensing. Exporting symbols. Writing kernel modules for extending Linux kernel.	
UNIT 4: GUI programming	14 Hrs.
X Window System - Introduction, history, features and working. X-Server, X-Protocol, X-Client, & X-lib. Qt toolkit – Introduction, cross-platform GUI development. Qt creator. Basic structure of a Qt program. Compilation. Signal-Slot mechanism. Qt widgets. Container widgets. Custom layouts and slots. Writing Qt programs for developing basic GUI applications.	

Textbooks:

1. R. Petersen, LINUX: The Complete Reference, 6th Edition, Tata McGraw Hill, 2008.

To be effective from year-2024

Reference Books:

1. S. Veeraraghavan. Shell Programming in 24 hours. SAMS/Techmedia, 2007.
2. R. Love. Linux Kernel Development. Addison-Wesley, 2010.
3. J. Blanchette, M. Summerfield. C++ GUI Programming with Qt3. Prentice Hall, 2004.

COURSE OUTCOMES (CO):

CO1: Students will be able to describe the structure, features and utilities available in Linux
CO2: Students will be able to use Linux utilities for system administration
CO3: Students will be able to develop basic applications using Shell scripting
CO4: Students will be able to describe various methods of extending a Linux kernel
CO5: Students will be able to develop kernel modules for extending Linux kernel
CO6: Students will be able to develop GUI applications using Qt programming

LEVEL OF CO-PO MAPPING TABLE

COs	Pos											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3	-	-	-	3	-	2	3	-	-	-	-
2	3	-	3	2	3	2	2	3	-	-	-	2
3	3	2	3	2	3	2	2	-	-	-	-	2
4	3	-	2	-	2	-	2	-	-	-	-	-
5	2	2	3	2	3	2	2	-	3	2	2	2
6	2	2	3	2	3	2	2	-	3	2	2	2

COURSE TITLE: Theory of Computation

Course Code: MCA24211DCE						Examination Scheme	
Total number of Lecture Hours: 56						External	80
						Internal	20
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits	4

Course Objectives:

- To Understand computational models and finite automata in formal language theory and computational complexity.
- To Design and analyze DFA and NFA, understand regular languages, and their equivalence with regular expressions.
- To Study context-free languages (CFLs), grammars (CFGs), parse trees, and pushdown automata (PDA).
- To Explore context-sensitive languages (CSL), linear bounded automata (LBA), recursive languages (REL), and Turing machines (TM).
- To Learn about decidability, undecidability, reduction techniques, and complexity theory foundations.

COURSE CONTENT	TEACHING HOURS
UNIT 1: Introduction to Computation	14 Hrs.
Introduction to computation, Regular Languages: Introduction to formal languages, regular operations, Closure property. Finite Automata, Deterministic Finite Automata, Kleene's theorem, Non-deterministic Finite Automata (NFA), ε-NFA, Conversion of ε-NFA to NFA, NFA to DFA, Minimization, Finite Automata with output: Mealy and Moore machines. Regular Expression; Equivalence of DFA, NFA, and RE. Non-Regular Languages and Pumping Lemma.	
UNIT 2: Context-Free Languages	14 Hrs.
Introduction to Context-Free Languages (CFL), Pushdown Automata (PDA), Grammars, Context Free Grammars, Parsing and Ambiguity, Parsing and Membership, Inherent Ambiguity of Context-Free Languages, Chomsky Normal Form, Membership Algorithm for CFG. Deterministic vs non-deterministic PDAs. Closure property and Pumping Lemma for CFLs.	
UNIT 3: Context-Sensitive Languages and Turing Machine	14 Hrs.
Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Languages (CSL), Context Sensitive Grammars, Linear Bounded Automata (LBA). Introduction to Turing Machines, Turing Machines as Language Acceptors and Transducers, Turing's Thesis, Equivalence of Deterministic, Non-deterministic, and multi-tape TMs. Universal TMs.	
UNIT 4: Undecidability and Computational Complexity	14 Hrs.
Decidability and Undecidability, Reductions and its applications, Rice's theorems for RE sets, Post Correspondence Problem. Halting Problem, Halting vs Looping. Hilbert's algorithm. Complexity Classes (P and NP), Satisfiability (SAT) Problem, Hamiltonian Path Problem, Clique Problem. Polynomial Time Reduction.	

To be effective from year-2024