Post Graduate Department of Computer Sciences,
University of Kashmir,
Srinagar-190006

Syllabus of One/Two Year MCA Programme
Effective from Year/Batch 2025

Eligibility Criteria

- **For Two-Year MCA: -** “Any Graduate with at least 12 credits in Computer Science / applications under CBCS/NEP 2020 Scheme

  OR

  B.Sc. with Mathematics (at 10+2 level OR in graduation) or B.Tech/BE"

- **For One-Year MCA:** "Any 4-Year graduate with at least 20 credits in Computer Science/applications or any other computing field".

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: To prepare students to get employment, profession and/or to pursue post-graduation and research in Computer Applications discipline in particular and allied Computer Science fields in general.

PEO2: To prepare students to identify and analyze problems in the computing perspective an develop computer applications solutions using an iterative approach that involves defining, designing, quantifying, implementing, testing, deploying and review of the solution to the problem..

PEO3: To prepare students to plan, organize, schedule, execute and communicate effectively as an individual, a team member or a leader in problem solving environment.

PEO4: To provide to students, an academic environment that makes them aware of excellence in field of Computer Sciences in general and enables them to understand significance of lifelong learning in global perspective.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Masters should be creative, imaginative and proficient software engineers employable to serve in the industry, government and allied services.

PSO2: Masters should be able to advance in academic and research pursuits in computing and allied disciplines.

PSO3: Masters should take a lead in innovation and entrepreneurship activities with high standards of professional and moral ethics and prove themselves beneficial to society at large.

Programme Learning Outcomes (PLOs)

1. **Knowledge and Understanding:**
   Evaluate core and advanced concepts in computing such as algorithms, system design, artificial intelligence, and data-driven systems, with the ability to interpret emerging trends and technologies.
2. **Technical Skills:**
   Design, implement, and evaluate software systems using modern computing tools, programming languages, and best practices in system development and deployment.
3. **Application of Knowledge and Skills:**
   Integrate computational knowledge with domain-specific problems to build and deploy practical solutions across sectors such as healthcare, environment, education, and business.
4. **Communication Skills:**
   Demonstrate the ability to effectively articulate technical ideas, research findings, and project outcomes in written, oral, and visual formats for diverse stakeholders.
5. **Critical Thinking:**
   Analyze complex technical problems, synthesize information from multiple sources, and apply logical reasoning to develop, test, and refine effective solutions.
6. **Ethics:**
   Apply ethical principles, legal guidelines, and professional standards in the design, implementation, and evaluation of computing technologies.

7. **Life-long Learning:**
   Pursue continuous professional growth and knowledge enhancement through research, certifications, conferences, and scholarly engagement.
8. **Creativity:**
   Design innovative computing models and user-centric solutions by applying creativity in algorithm development, interface design, and system architecture.
9. **Research Aptitude:**
   Formulate research problems, conduct literature reviews, design experimental setups, and evaluate results using appropriate scientific methods and tools.
10. **Problem Solving:**
    Identify real-world challenges, define computational problems clearly, and develop optimized, scalable solutions using algorithmic thinking and systematic evaluation.

Two Year MCA Syllabus Structure (CW+R)

Credit Level	Semester	Course Type	Course Code with Name	Course Level	Credits	Total Credits	Max. Marks			Credit Distribution	Contact Hours
							Continuous Assessment	End Semester	Total	L: T: P	
6.0	Sem - I	Core	MMCACJP125: Java Programming	400	4	22	28	72	100	4:0:0	60
		Core	MMCACML125: Machine Learning	400	4		28	72	100	4:0:0	60
		DCE-I	MMCADAD125: Advanced Data Structures	400	4		28	72	100	4:0:0	60
			MMCADCG125: Computer Graphics								
			MMCADMI125: Management Information System								
			MMCADSE125: Software Engineering								
		DCE-II	MMCADDS125: Advanced Database Systems	400	4		28	72	100	4:0:0	60
			MMCADAI125: Artificial Intelligence								
			MMCADBC125: Block Chain Technologies								
			MMCADCS125: Cyber Security & Digital Forensics								
		Core	MMCACRM125: Research Methodology	400	2		14	36	50	2:0:0	30
		Lab	MMCALJP125: Java Programming Lab	400	2		14	36	50	0:0:2	60
		Lab	MMCALML125: Machine Learning Lab	400	2		14	36	50	0:0:2	60
	Sem - II	Core	MMCACDA225: Design and Analysis of Algorithms	400	4	22	28	72	100	4:0:0	60
		Core	MMCACMA225: Mobile Application Development	400	4		28	72	100	4:0:0	60
		DCE-III	MMCADAO225: Advanced Operating Systems	400	4		28	72	100	4:0:0	60
			MMCADDI225: Digital Image Processing								
			MMCADDS225: Decision Support Systems								
			MMCADCN225: Cryptography & Network Security								
		DCE-IV	MMCADAC225: Advanced Computer Networks	400	4		28	72	100	4:0:0	60
			MMCADCC225: Cloud Computing								
			MMCADLP225: Linux Programming								
			MMCADTC225: Theory of Computation								
		Core	MMCACRP225: Research and Publication Ethics	400	4		28	72	100	4:0:0	60
		Lab	MMCALMA225: Mobile Application Development Lab	400	2		14	36	50	0:0:2	60
Total (First Year)					**44**	**44**	**308**	**792**	**1100**	**38:0:6**	**750 Hrs**
6.5	Sem - III	Core	MMCACDS325: Data Science with Python	500	4	22	28	72	100	4:0:0	60
		Core	MMCACWP325: Web Programming	500	4		28	72	100	4:0:0	60
		DCE-V	MMCADQC325: Quantum Computing	500	4		28	72	100	4:0:0	60
			MMCADEH325: Ethical Hacking								
			MMCADCV325: Computer Vision								
			MMCADER325: Enterprise Resource Planning								
		DCE-VI	MMCADNL325: Natural Language Processing	500	4		28	72	100	4:0:0	60
			MMCADSQ325: Software Quality Assurance								
			MMCADDL325: Deep Learning								
			MMCADIT325: Internet of Things								
		Core	MMCACSP325: Software Project Management	500	2		14	36	50	2:0:0	30
		Lab	MMCALDS325: Data Science with Python Lab	500	2		14	36	50	0:0:2	60
		Lab	MMCALWP325: Web Programming Lab	500	2		14	36	50	0:0:2	60
	Sem - IV	Project	MMCAPPI425: Problem Identification & Analysis	500	6	20	42	108	150	6:0:0	90
		Project	MMCAPDI425: Dissertation	500	6		42	108	150	6:0:0	90
		Project	MMCAPSD425: Software Development	500	4		28	72	100	0:0:4	120
		Project	MMCAPRC425: Research Component	500	4		28	72	100	0:0:4	120
Total (Second Year)					**42**	**42**	**294**	**756**	**1050**	**30:0:12**	**810 Hrs**
TOTAL CREDITS (AGGREGATE OF 4-SEMESTERS)					***86***	***86***	***602***	***1548***	***2150***	***68:0:22***	***156 0 Hrs***

SEMESTER - I

COURSE TITLE:JAVA PROGRAMMING

Course Code: MMCACJP125		**Examination Scheme**	
Total number of Lecture Hours: 60		**External**	**72**
		Internal	**28**

Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**

Course Objectives

- To understand the fundamental principles of Java programming language, including its syntax, semantics, and basic constructs.
- To explore object-oriented programming concepts such as classes, inheritance, polymorphism, and interfaces in the context of Java.
- To develop proficiency in handling exceptions and errors using Java's exception handling mechanisms.
- To gain practical experience in utilizing Java's standard library classes and packages for tasks like I/O operations, string manipulation, and multithreading.
- To learn to create graphical user interfaces (GUIs) in Java, employing event-driven programming paradigms and integrating various GUI elements.
- To acquire skills in network programming with Java, including socket programming for communication between distributed systems and applications.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Java Programming	**15- Hrs**
Introduction to Java Language: Creation of Java. How Java changed the Internet. Features of Java Language. Evolution of Java. Comparison with other languages like C++.Java Virtual Machine (JVM) and Byte-code. Java Language Overview: Lexical issues – Whitespace, Identifiers, Keywords, Literals, Separators, and Comments. Installing JDK.PATH variable. Java program – Structure, Compilation and Execution. Java Class libraries (System Class).main() method. **Data types, Variables and Arrays:** Primitive Data-types and Typed-Literals. Variables – Declaration, Initialization, Scope and Lifetime. Arrays – Single and Multidimensional. Type Conversion and Expression Promotion. **Operators, Expressions and Control statements:** Arithmetic, Bitwise, Relational, Logical, Assignment. Precedence and Associativity. Selection, Iteration and Jump Statements.	
UNIT 2: Object-Oriented Programming in Java	**15- Hrs**
Class Fundamentals: Class Structure (Variable and Method declaration).Modifiers (Access Modifiers and Other Modifiers).Components of Class, Variable and Method declaration. Constructor and finalize(). Garbage Collection. Passing parameters to methods. Variable hiding. Method overloading. Constructor overloading and chaining. Use of this keyword. Code blocks - Static and non-static. **Inheritance:** Mechanism. Role of Access Modifiers. Method Overriding and Shadowing. Use of super keyword. Polymorphism - Early and Late binding. Abstract Class and Interface. Components of Interface declaration. Implementing Interfaces. **Exception Handling:** Mechanism - Exception-Object, Throwing an Exception, and Exception Handler. Catch or Specify policy. Types of Exception - Checked vs Unchecked, Built-in vs Userdefined. Catching an Exception - try-catch-finally.	

Specifying an Exception - throws. Manually throwing an Exception - throw. Custom Exceptions. Chained Exceptions.

UNIT 3: Advanced Java Concepts	15- Hrs
Packages: Creating and Importing Packages. CLASSPATH variable. static import. **Strings:** Mutable and Immutable Strings. Creating Strings. Operations on Strings. **Threads:** Creating Threads in Java. Java Thread Lifecycle. Multithreading in Java: Synchronization and Inter- process communication (IPC) in Threads. **I/O Streams:** Byte, Character, Buffered, Data, and Object Streams. Standard Streams. File I/O Basics, Reading and Writing to Files. Serializing Objects.	

UNIT 4: Java GUI Programming and Networking	15-Hrs
Event-Driven Programming: Java 1.1 Event Delegation Model – Source object, Event object and Listener object. Methods associated with Source, Event and Listener objects. Low-level vs Semantic events. Adapter classes, Inner classes, and Anonymous Inner classes. Adding GUI elements to Applet. **Networking Classes and Interfaces:** TCP/IP Server Sockets in Java. Developing simple networking applications in Java like File transfer, Chatting, etc.	

Textbooks

1. H. Schildt, Java: The Complete Reference, 13th Edition, Tata McGraw Hill, 2023.

Reference Books

1. E. Balagurusamy, Programming with Java: A Primer, 7th Edition, Tata Mcgraw Hill, 2023.
2. H.M. Dietel and P.J. Dietel, Java: How to Program, 11th Edition, Pearson Education, 2017.
3. K. Sierra and B. Bates, Head First Java (Java 5), 2nd Edition, O'Reilly, 2003.
4. C.S. Horstmann and G. Cornell, Java 2 Vol-1 Fundamentals, 7th Indian Reprint, Pearson Education, 2006.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understanding of the foundational concepts of Java programming, including data types, control structures, program flow, and compilation/execution of Java applications.

CLO2: Apply object-oriented programming principles in Java using classes, inheritance, polymorphism, interfaces, and exception handling for robust application development.

CLO3: Demonstrate the use of advanced Java features such as multithreading, string manipulation, package management, and file I/O operations.

CLO4: Design and implement event-driven GUI applications and basic networking solutions using Java APIs and socket programming.

CLOs	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	Average (CLO)
MMCACJP125.1	3	2	1	1	2	1	1	0	2	2	1.5
MMCACJP125.2	2	3	2	1	3	1	1	2	1	3	1.9
MMCACJP125.3	2	3	3	1	3	1	2	2	1	3	2.1
MMCACJP125.4	2	3	3	2	3	1	2	3	1	3	2.3
Average(PLO)	2.25	2.75	2.25	1.25	2.75	1	1.5	1.75	1.25	2.75	1.95

COURSE TITLE: Machine Learning

Course Code: MMCACML125						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To introduce the fundamental concepts, techniques, and applications of machine learning and provide insight into its challenges and testing methods.
- To equip students with the ability to build, evaluate, and optimize basic machine learning models including regression and classification models.
- To develop proficiency in various clustering techniques and feature engineering for unsupervised learning scenarios.
- To enable students to apply advanced classification methods such as Bayesian learning and Support Vector Machines for solving complex real-world problems.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Machine Learning and Data Preprocessing	**15 Hrs.**
Machine Learning, Applications, Types of Learning, Main Challenges of Machine Learning, Testing and Validating, designing a learning System, Inductive Bias and Hypothesis, Hypothesis Evaluation, Feature extraction, Types of feature selection, Feature Handling, Normalization, Missing data, Dimensionality Reduction: Principal Component Analysis	
UNIT 2: Regression, Classification, and Clustering Basics	**15 Hrs.**
Linear Regression, Logistic Regression, Decision Tree Representation, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Learning, Inductive Bias in Decision Tree Learning, Issues in Decision Tree Learning, Clustering Algorithms: Euclidean and Mahalanobis Distance, K-means algorithm	
UNIT 3: Advanced Clustering Techniques and Instance-Based Learning	**15Hrs.**
Cluster validity index, Compactness Cluster measure, Distinctness Cluster Measure, Fuzzy C-means, Hierarchical Clustering, Density based spatial clustering of applications with noise (DBSCAN), Spectral clustering, k-medoids clustering, Kohenon Self Organizing Net, K- Nearest Neighbour and effect of various Distance measures	
UNIT 4: Probabilistic Learning and Support Vector Machines	**15 Hrs.**
Bayesian Learning: (Bayes Theorem and Concept Learning, Maximum Likelihood and Least- Squared Error Hypothesis, Naïve Bayes Classifier, Bayesian Belief Networks). Support Vector Machine: Linear Support Vector Machine, Optimal Hyperplane, Kernel functions, Solving Non-Linear Classification problems with Linear Classifier, Multiclass Support Vector Machines, Applications of Support Vector	

Machines	

Textbooks

1. Machine Learning by Tom M. Mitchel, McGraw-Hill publication

Reference Books

1. Pattern Classification by Duda and Hart. John Wiley publication
2. The Elements of Statistical Learning: Data Mining, Inference, and Prediction by Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer.
3. Learning From Data, Yaser S. Abu-Mostafa, Hsuan-Tien Lin, Malik Magdon-Ismail, AML Book.
4. Introduction to Machine Learning by EthemAlpaydin, The MIT Press.
5. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Chapman and Hall/CRC.

COURSE LEARNING OUTCOMES (CO):

After completing the course, the student will be able to:

CLO1: Understand the types, challenges, and foundations of machine learning, and apply preprocessing techniques such as feature selection, normalization, and dimensionality reduction.
CLO2: Implement and analyze basic supervised and unsupervised machine learning algorithms including regression, decision trees, and k-means clustering.
CLO3: Apply and evaluate advanced clustering techniques and instance-based learning models for complex pattern discovery.
CLO4: Implement and compare probabilistic models and support vector machines for classification and prediction in various application domains.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCACML125.1	3	2	2	2	2	1	2	2	2	2	**2.0**
MMCACML125.2	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCACML125.3	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCACML125.4	3	3	3	2	3	1	2	2	2	3	**2.4**
Average (PLO)	**3.0**	**2.75**	**2.75**	**2.0**	**2.75**	**1.0**	**2.0**	**2.0**	**2.0**	**2.75**	**2.3**

DCE-I

COURSE TITLE: Advanced Data Structures

Course Code: MMCADAD125	**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60	**External**	**72**	**-**
	Internal	**28**	**-**

Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Objectives

- To understand and implement fundamental linear data structures and algorithms, including arrays, searching, sorting, matrices, and linked lists.
- To apply stack and queue data structures to solve computational problems using array and linked list implementations.
- To analyze and implement tree and graph structures along with their traversal techniques and real-world applications.
- To explore advanced data structures and algorithms including hashing, heaps, and file organizations for optimized data processing.

Course Content	**TEACHING HOURS**
Unit I: Linear Data Structures	**15 Hrs.**
Data types/objects/structures, Data structures and its types, Representation and implementation. Linear Data Structures: Array representation, operations, applications and limitations of linear arrays, Searching Techniques- Linear Search, Binary Search Sorting Techniques- Selection, Insertion sort, Bubble sort, Quick Sort, Merge Sort Two dimensional arrays, matrices, common operations of matrices, special matrices, Array representation of Sparse matrices. Linked Lists: Representation, Types and operations on Linked List.	
Unit II: Stack and Queues	**15 Hrs.**
Stack- Representation of stack in memory, Operations on Stacks, Implementation of Stack using arrays and linked list, Multiple Stacks: Representing two stacks and more than two stacks, Applications of stacks: Parenthesis Checker, Infix to postfix procedure, evaluating expressions in postfix notation, Implementation of recursion using stack. Queues- Representation of Queue in Memory, Operations on Queue, Implementation of Queue using arrays and linked list, Circular Queue and its operations, Representation and implementation, Multiple Queues, Deque, Priority Queue, Heap Representation of a Priority Queue, Applications of Queues.	
Unit III: Tree and Graph Data Structures	**15 Hrs.**
Trees, Definitions, terminologies and properties, Binary tree representation, traversals and applications, Threaded binary trees, Binary Search Trees, AVL Trees, M-way Search Trees, B-trees, B+ trees. Graphs, Terminology, Graph representations, Traversal Techniques, Operations on Graphs, Applications of Graphs	
Unit IV: Advanced Data Structures and Algorithms	**15 Hrs.**

Minimum spanning trees, Shortest Path Algorithms in Graphs, Eulerian Tour, Hamiltonian Tour Hashing: Direct Address Tables, Hash Table, Different Hash functions, resolving collisions, rehashing, Heap Structures, Binomial Heaps, Leftist Heaps. File Organizations: Sequential File Organization, Relative File Organization, Indexed Sequential File Organization, Multiple Key File Organizations: Inverted File and Multi-List Organizations	

Textbooks

1. Langsam, Augenstein, Tenenbaum, “Data Structures Using C and C++”, 2nd Edition, 2015

Reference Books

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson Freed, “Fundamentals of Data Structures In C”, 2nd Edition, 2018
2. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C++”, 3rd Edition, 2007.
3. Aho Alfred V., Hopcroft John E., UIlman Jeffrey D, “Data Structures and Algorithms”, 2017
4. R. S. Salaria, “Data Structures and Algorithms Using C++”, 2018
5. Varsha H Patil, “Data Structures using C++”, 2012
6. E.Balagurusamy, “Object Oriented Programming with C++” , 8th Edition, 2020

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand and implement fundamental linear data structures and algorithms, including arrays, searching, sorting, matrices, and linked lists.
CLO2: Apply stack and queue data structures to solve computational problems using array and linked list implementations.
CLO3: Analyze and implement tree and graph structures along with their traversal techniques and real-world applications.
CLO4: Explore advanced data structures and algorithms including hashing, heaps, and file organizations for optimized data processing.

CLO-PLO Matrix for the Course

	PLOs										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average (CLO)**
MMCADAD125.1	3	3	2	1	3	0	2	2	1	3	**2.0**
MMCADAD125.2	3	3	2	1	3	0	2	2	1	3	**2.0**
MMCADAD125.3	3	3	3	1	3	0	2	3	2	3	**2.3**
MMCADAD125.4	3	3	3	1	3	1	2	3	2	3	**2.4**
Average (PLO)	**3.0**	**3.0**	**2.5**	**1.0**	**3.0**	**0.25**	**2.0**	**2.5**	**1.5**	**3**	**2.2**

COURSE TITLE: Computer Graphics

Course Code: MMCADCG125	**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60	**External**	**72**	**-**
	Internal	**28**	**-**

Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Objectives:

- To understand key concepts, graphic display devices, and 2D/3D transformations.
- To develop skills in line and circle drawing, clipping, filling, and hidden surface removal.
- To apply mathematical techniques like splines and Bezier methods for complex graphical models.
- To explore multimedia concepts, file formats, storage solutions, and introductory AR/VR technologies**.**

Course Content	**TEACHING HOURS**
UNIT 1:	**15Hrs**
Introduction to Computer Graphics, Applications of Computer Graphics, Graphic Display Devices: Refresh Cathode Ray Tubes, Raster-scan Displays, Random-Scan displays, Color CRT Monitors, Concept of Double Buffering, Lookup tables. 2-D Graphics: Cartesian and Homogeneous Coordinate Systems, Line drawing algorithms (Bressenham's and DDA), Circle and Ellipse Drawing Algorithms.	
UNIT 2:	**15Hrs**
2-Dimensional Transformations, Concepts of Window & Viewport, Window to Viewport Transformations, Normalization transformation (3L) Composite Transformations: General pivot point rotation, General fixed point scaling, reflection w.r.t line y=x, reflection w.r.t line y=x (4L) Transformation between coordinate systems, affine transformations, Raster methods for transformations (3L)	
UNIT 3:	**15Hrs**
Filling techniques: Boundary and Flood-fill algorithms (2L) Clipping, Line Clipping Algorithms (Cohen-Sutherland Algorithm), 3-D Graphics, Projections: perspective and parallel projection transformations. (5L) 3-Dimensional Transformations, Hidden Surface Removal Techniques, Z-Buffer Algorithm, Back Face Detection (3L)	
UNIT 4:	**15Hrs**
Curves and Surfaces: Spline specification, Interpolated& Approximated Splines. spline representation, cubic spline interpolation methods, Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. (3LIntroduction to multimedia elements: Images (BMP, PCX), sound (WAV, MP3) Multimedia storage formats: CDs and DVDs). Introduction to virtual reality (VR) and augmented reality (AR) technologies.	

Textbooks

1. Hearn and Baker, "Computer Graphics with OpenGL": 4th Edition (2022), Donald Hearn, M. Pauline Baker, Warren Carithers, Pearson
2. Ze-Nian Li and Mark S. Drew, "Fundamentals of Multimedia": 3rd Edition (2021), Springer.

3. **W.M. Newman and Sproull, “Principles of Interactive Computer Graphics”** McGraw-Hill Education; 3rd Edition, 2023.

Reference Books

1 **Steven Harrington, “Computer Graphics: A Programming Approach”** McGraw-Hill Education; 2nd Edition, 2021.
2 **Plastock and Kelley, “Schaum's Outline of Theory and Problems of Computer Graphics”** McGraw-Hill Education; 2nd Edition, 2022.
3 **David F. Rogers and J. Alan Adams, “Procedural Elements of Computer Graphics”** McGraw-Hill Education; 3rd Edition, 2021.
4 **David F. Rogers and J. Alan Adams, “Mathematical Elements of Computer Graphics”** McGraw-Hill Education; 3rd Edition, 2022.
5 **James D. Foley, Andries van Dam, et al., “Computer Graphics: Principles and Practice”** Pearson; 4th Edition, 2023.
6 **Sinha and Udai, “Computer Graphics”** Tata McGraw-Hill Education; 2nd Edition, 2022

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand basic computer graphics concepts, display devices, and 2-D drawing algorithms.
CLO2: Apply 2-D transformations, window-to-viewport mapping, and coordinate conversions.
CLO3: Implement filling, clipping algorithms, 3-D projections, and hidden surface removal.
CLO4: Analyze spline curves, multimedia basics, and introduction to VR/AR technologies.

LEVEL OF CO-PO MAPPING TABLE

	PLO										
UNIT-WISE CLOs	1	2	3	4	5	6	7	8	9	10	**Avg (CLO)**
MMCADCG125.1	3	2	0	0	0	0	0	0	0	0	**2.5**
MMCADCG125.2	0	3	3	2	0	0	0	0	0	0	**2.67**
MMCADCG125.3	0	0	0	0	3	3	0	0	0	0	**3.0**
MMCADCG125.4	0	0	0	0	0	0	3	3	0	0	**3.0**
Avg (PLO)	**1.5**	**1.67**	**1.5**	**1.0**	**1.5**	**1.5**	**1.5**	**1.5**	**0**	**0**	**2.79**

COURSE TITLE: Management Information System

Course Code: MMCADMI125						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- To understand the structure of organizations and the role of various information systems (MIS, DSS, GDSS).
- To analyze system requirements using structured system analysis tools and methods.
- To Explore enterprise systems like ERP, SCM, and CRM, and their role in strategic IT decisions.
- To ethical Evaluate the ethical, security, and social issues surrounding the use of information systems.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Organizations and Information Systems	**15 Hrs.**
Organization and Information Systems, The Organization: Structure, Managers and activities – Data ,information and its attributes – The level of people and their information needs - Types of Decisions and information - Information System, - Management Information System (MIS) –Decision Support System (DSS) and Group Decision Support System (GDSS).	
UNIT 2: System Analysis and Development	**15 Hrs.**
Need for System Analysis - Stages in System Analysis - Structured SAD and tools like DFD, Context level Diagram, Decision Table and Structured Diagram. System Development Models: Waterfall, Prototype, Spiral, –Roles and responsibilities of System Analyst, Database Administrator and Database Designer.	
UNIT 3: Enterprise Systems and IT Decision-Making	**15 Hrs.**
Enterprise Resources Planning (ERP): Features, selection criteria, merits, issues and challenges in Implementation - Supply Chain Management (SCM): Features, Modules in SCM – Customer Relationship Management (CRM): Phases. Knowledge Management and e-governance, Nature of IT decisions- Strategic decision.	
UNIT 4: Security, Ethics, and Social Challenges in Information Systems	**15 Hrs.**
Security and Ethical Challenges, Ethical responsibilities of Business Professionals – Business, technology, Computer crime – Hacking, cyber theft, unauthorized use at work. Issues and internet privacy. Challenges – working condition, health and social issues, Ergonomics and cyber terrorism.	

Textbooks

1. 1. "Management Information Systems", Kenneth J Laudon, Jane P. Laudon, Pearson/PHI,10/e, 2007
2. "Management Information Systems", W. S. Jawadekar, Tata McGraw Hill Edition, 3/e, 2004

Reference Books: -

1. Turban, Efraim, Ephraim McLean, and James Wetherbe. 2007. Information Technology for Management: Transforming Organizations in the Digital Economy. New York, John Wiley & Sons.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Describe the organizational structure and classify different types of information systems based on managerial needs.
CLO2: Apply system analysis and design techniques including DFDs, context diagrams, and decision tables.
CLO3: Evaluate and compare enterprise systems such as ERP, SCM, and CRM and understand IT-based strategic decisions.
CLO4: Analyze ethical, legal, and security issues in the management and use of information systems.

LEVEL OF CO-PO MAPPING TABLE

Unit wise CLOs	PLOs										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADMI125.1	3	2	2	2	2	2	1	2	1	2	1.9
MMCADMI125.2	2	3	3	3	3	1	1	1	2	2	2.1
MMCADMI125.3	2	2	2	2	2	3	2	2	2	3	2.2
MMCADMI125.4	2	2	2	2	2	3	2	3	2	3	2.3
Avg (PLO)	**2.25**	**2.25**	**2.25**	**2.25**	**2.25**	**2.25**	**1.5**	**2.0**	**1.75**	**2.5**	**2.1**

COURSE TITLE: Software Engineering			
Course Code: MMCADSE125	**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60	**External**	**72**	**-**
	Internal	**28**	**-**

Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**

Course Objectives

- To Gain knowledge of the nature, goals, and challenges of software engineering and its historical context.
- To Learn and utilize various software development models, including Waterfall, Agile, and Spiral.
- To Analyze software processes using measures, metrics, and models like CMMI and COCOMO.
- To Develop skills in eliciting, analyzing, modeling, and validating both functional and non-functional requirements.
- To Understand design principles, modularity, and patterns, and apply function-oriented and object-oriented design methodologies.
- To Understand core testing concepts and techniques, and explore software reliability and reengineering processes.

Course Content	**TEACHING HOURS**
UNIT 1: Fundamentals of Software Engineering	**15 Hrs**
Concept and Nature of Software: Concept and Nature of Software, Software Crisis, Software Engineering – Concept, Goals and Challenges, Software Engineering Approach. Software Development Process, Process Models - Waterfall Model, Evolutionary and Throwaway Prototyping Model, Incremental and Iterative Models, Spiral Model, Agile Process Model, Component based and Aspect Oriented development Software Process and Project Measurement: Measures, Metrics and Indicators, Size - Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model.	
UNIT 2: Requirements Engineering	**15 Hrs**
Introduction to Requirements Engineering - Why, What and Where. Requirements Types: functional and nonfunctional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques. Requirement Analysis and Modelling, Requirements prioritization, verification, and validation.	
UNIT 3: Design Engineering	**15 Hrs**
Basics of Design Engineering - Abstraction, Architecture, Patterns, Separation of concerns, Modularity, Functional Independence, refinement, Refactoring. Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology. Object-Oriented Design - Design Concepts, Design Methodology, Object-oriented analysis and design modeling using Unified Modeling Language (UML), Dynamic & Functional Modeling, Design Verification.	

UNIT 4: Software Testing and Reliability	15 Hrs
Software Testing – Concepts, Terminology, Testing & Debugging, Adequacy Criteria, Static vs. Dynamic Testing, Black Box vs. White Box Testing. Structural testing and its techniques. Functional Testing and its techniques, Mutation testing, Random Testing. Non-Functional Testing like Reliability, Usability, Performance and Security Testing. Introduction to Software Reliability: Basic Concepts, Correctness Vs Reliability, Software Reliability metrics, Operational Profile, Reliability Estimation and Predication, Reliability and Testing. Concept of Software reengineering, reverse engineering and change management.	

Textbooks

1. Shari Lawrence Pfleeger and Joanne M. Atlee - "Software Engineering: Theory and Practice," 4th Edition, Pearson, 2010.

Reference Books

1. Ian Sommerville - "Software Engineering," 10th Edition, Pearson, 2015.
2. Pankaj Jalote - "An Integrated Approach to Software Engineering," 3rd Edition, Narosa Publishing House, 2005.
3. Hans Van Vliet - "Software Engineering: Principles and Practice," 4th Edition, Wiley, 2016.
4. James F. Peters - "Software Engineering: An Engineering Approach," 1st Edition, Wiley & Sons, 2000.
5. Roger Pressman - "Software Engineering: A Practitioner's Approach," 8th Edition, McGraw-Hill Publications, 2014.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand software engineering concepts, process models, and measurement techniques, and apply these to estimate and plan software development projects.
CLO2: Identify and analyze functional and non-functional requirements using requirement engineering processes and frameworks.
CLO3: Apply principles and practices of software design including structured and object-oriented approaches using UML and software design methodologies.
CLO4: Analyze and apply software testing techniques and reliability metrics for validating and verifying software quality and performance.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCADSE125.1	3	3	3	2	2	1	2	2	2	3	**2.3**
MMCADSE125.2	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCADSE125.3	3	3	3	2	3	1	2	3	2	3	**2.5**
MMCADSE125.4	3	3	3	2	3	1	2	2	2	3	**2.4**
Average (PLO)	**3.0**	**3.0**	**3.0**	**2.0**	**2.75**	**1.0**	**2.0**	**2.25**	**2.0**	**3.0**	**2.4**

DCE-II

COURSE TITLE: Advanced Database Systems				
Course Code: MMCADDS125	**Examination Scheme**	**T**		**P**
Total number of Lecture Hours: 60	**External**	**72**		**-**
	Internal	**28**		**-**

Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To understand the foundational concepts of database systems, including data models, architecture, and ER modeling, in order to distinguish between data, information, and knowledge.
- To explore the relational data model and relational algebra operations, and to apply normalization techniques for designing efficient and consistent database schemas.
- To gain a conceptual understanding of distributed, parallel, and object-based database systems, their architectures, key features, and differences from traditional database models.
- To examine the principles of transaction management and recovery, including concurrency control techniques, serializability, and recovery methods to ensure database integrity and reliability.

Course Content	**TEACHING HOURS**
UNIT I: Introduction to Database Systems	**15 Hrs**
Introduction to Data, Information and Knowledge. Database basics – Need and evolution, Database and DBMS. Characteristics of Database Approach, Advantages and disadvantages of DBMS Approach. Database System Concepts and Architecture – Data Models, Schemas, and Instances, Database Models and Comparison, Three Schema Architecture and Data Independence. Database Languages and Interfaces. DBMS architectures. DBMS Classification. Data Modeling: Overview of Data Modeling, Entity-Relationship (ER) Modeling.	
UNIT II: Relational Data Model and Database Design	**15 Hrs**
Relational Data Model – Basic Concepts and Characteristics, Model Notation, Model Constraints and Database Schemas, Constraint Violations. Relational Algebra – basic concepts, Unary Relational Operations, Algebra Operations from Set Theory, Binary Operations, Additional Relational Operations. Criterion for Good Database Design. Database Design through Functional Dependencies & Normalization: Functional Dependencies, Lossless Join, Normal Forms: 1NF, 2NF, 3NF, BCNF.	
UNIT III: Overview of Distributed, Parallel, and Object-Based Databases	**15 Hrs**
Distributed Databases – Basic Concepts, Characteristics, and Design Issues, Data Fragmentation and Replication , Distributed Transparency: Location, Replication, and Fragmentation Transparency. **Parallel Databases** – Architecture Types (Shared Memory, Shared Disk, Shared Nothing), Concepts of Parallel Query Processing and Optimization. **Object-Based Databases** – Motivation, Features, and Architecture, Concepts of Object-Oriented Data Models and Object-Relational Databases, Comparison with Traditional Relational Databases	

UNIT IV: Transaction Management and Recovery	**15 Hrs**
Concepts of Transactions – ACID Properties, Transaction States, Issues in Concurrency – Lost Updates, Dirty Reads (Conceptual Examples), Serializability – Conflict and View Serializability (Overview Only), Overview of Concurrency Control Techniques – Two-Phase Locking, Timestamp Ordering, Database Recovery – Causes of Failure, Rollback Mechanisms, Recovery Approaches – Deferred and Immediate Update, Shadow Paging, Introduction to Logging and Checkpointing Techniques	

Textbooks:

1. Advanced Database Systems by Nabil R. Adam and Bharat K . Bhargava, ISBN 3 54057507-3 Springer-Verlag Berlin Heidelberg New York

Reference Books:

1. Ramez Elmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, 7thEdition, Pearson Education, 2017
2. Advanced Database Systems by Dr.John Kandiri
3. Abraham Silberschatz, Henry F. Korth, S.Sudarshan, “Database System Concepts”, 6th Edition, 2014

COURSE LEARNING OUTCOMES (CLO):

CLO1: Explain database architecture, data models, and the advantages of the DBMS approach in organizing and managing structured data.

CLO2: Apply relational algebra operations and normalization rules to evaluate, optimize, and design database schemas that preserve integrity and consistency.

CLO3: Differentiate between centralized, distributed, parallel, and object-based databases, and describe their architectures, transparencies, and processing capabilities.

CLO4: Interpret and evaluate transaction management strategies and recovery techniques, ensuring database consistency and reliability in multi-user environments.

CLO-PLO Matrix for the Course

	PLO										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADDS125.1	3	2	2	1	2	1	1	2	2	2	**1.8**
MMCADDS125.2	3	2	2	1	3	1	1	1	2	2	**1.8**
MMCADDS125.3	3	3	2	1	3	1	1	2	1	3	**2.0**
MMCADDS125.4	3	3	3	1	3	2	1	2	1	3	**2.2**
Avg (PLO)	**3.0**	**2.5**	**2.3**	**1.0**	**2.8**	**1.3**	**1.0**	**1.8**	**1.5**	**2.5**	**1.97**

<table>
<tr><th colspan="10">COURSE TITLE: Artificial Intelligence</th></tr>
<tr><td colspan="6"><b>Course Code: MMCADAI125</b></td><td><b>Examination Scheme</b></td><td colspan="2"><b>T</b></td><td><b>P</b></td></tr>
<tr><td colspan="6" rowspan="2"><b>Total number of Lecture Hours: 60</b></td><td><b>External</b></td><td colspan="2"><b>72</b></td><td><b>-</b></td></tr>
<tr><td><b>Internal</b></td><td colspan="2"><b>28</b></td><td><b>-</b></td></tr>
<tr><td><b>Lecture(L):</b></td><td><b>4</b></td><td><b>Practical (P):</b></td><td><b>-</b></td><td><b>Tutorial(T):</b></td><td></td><td><b>Total Credits</b></td><td colspan="3"><b>4</b></td></tr>
</table>

Course Objectives:

- To introduce the foundational concepts, historical evolution, knowledge representation methods, and intelligent agent models in Artificial Intelligence.
- To develop an understanding of fuzzy logic principles, inference techniques, and their application in handling imprecise or uncertain data.
- To equip students with various search and optimization techniques used in AI for problem-solving and decision-making.
- To explore inductive learning methods, uncertainty handling, and the fundamentals of artificial neural networks for pattern recognition and classification tasks.

Course Content	**TEACHING HOURS**
UNIT I:	**-15 Hrs**
Definition, history and applications AI. Representation of knowledge using logic-based approaches: propositional and first-order logic. Expert systems. Reasoning methods: Forward and backward chaining. Intelligent agents: types, structure, and environment interaction.	
UNIT II:	**- 15 Hrs**
Introduction to fuzzy logic and fuzzification. Linguistic terms, fuzzy sets, hedges, and operations. Fuzzy reasoning: Max-Min and Max-Product inferencing, multiple premise rules, Mamdani inference, aggregation, and defuzzification. Applications of fuzzy logic.	
UNIT III:	**-15 Hrs**
Search Algorithms – Uninformed search strategies, Informed search strategies, Hill Climbing, Constraint satisfaction problems, Optimization techniques: Genetic algorithms, Simulated annealing, Ant colony optimization, Swarm Particle optimization.	
UNIT IV:	**-15Hrs**
Inductive learning: categories, Rule extraction. Handling uncertainty in AI. Artificial Neural Networks (ANN): Basics of neural networks, architecture of perceptron and multilayer networks.	

Textbooks

1. *"***Artificial Intelligence: A Guide to Intelligent Systems"** by **Michael Negnevitsky, Latest Edition, 2020.**
2. *"Artificial Intelligence: A Modern Approach"* by Stuart Russell and Peter Norvig, **4th Edition,** 2020.
3. "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell, **Latest Edition,** 2019

Reference Books

1. "**Artificial Intelligence"** by **Elaine Rich, Kevin Knight, and Shivashankar B. Nair, 4th Edition, 2021**.
2. "Artificial Intelligence: Foundations of Computational Agents" by Michael Wooldridge, Ist Edition, 2021.
3. **"Nature-Inspired Optimization Algorithms"** by **Saeid Aziznejad, Gholamreza Z. Naderpour, and**

Mohammad A. H. Sadeghi, Ist Edition, 2019.

COURSE LEARNING OUTCOMES(CO):

CLO1: Identify and discuss various applications of AI across different domains and their impacts.

CLO2: Students will be able to explain the concepts of fuzzy logic including fuzzification and defuzzification.

CLO3: Implement and evaluate informed and uninformed search algorithms to solve problem-solving tasks.

CLO4: Students will be able to explain the principles of inductive learning and distinguish between different categories of inductive learning algorithms.

CLO-PLO Matrix for the Course

	PLO										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADAI125.1	3	1	2	2	2	1	1	2	1	2	**1.70**
MMCADAI125.2	3	2	2	1	2	1	1	2	1	2	**1.70**
MMCADAI125.3	3	2	2	1	2	1	1	1	1	2	**1.60**
MMCADAI125.4	2	3	3	1	3	1	2	3	2	3	**2.30**
Avg (PLO)	**2.75**	**2.00**	**2.25**	**1.25**	**2.25**	**1.00**	**1.25**	**2.00**	**1.25**	**2.25**	**1.82**

COURSE TITLE: Block Chain Technologies

Course Code: MMCADBC125						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**	

Course Objectives:

- To explain the foundational concepts of blockchain technology, including its structure, cryptographic principles, consensus mechanisms, and major blockchain platforms.
- To Analyze blockchain network components, security mechanisms, and real-world use cases, with a focus on the architecture and functioning of cryptocurrencies like Bitcoin and Ethereum.
- To Develop and deploy smart contracts and decentralized applications (DApps) using appropriate blockchain development tools, languages, and environments.
- To Evaluate emerging trends and interdisciplinary applications of blockchain in areas such as IoT, AI, big data, and quantum computing, and assess the potential of future technologies like Web 3.0 and decentralized identity..

Course Content	**TEACHING HOURS**
Unit 1: Introduction to Blockchain Technology	**15 Hrs**
Introduction to Blockchain - Definition, History, and Evolution. Basic Concepts - Distributed Ledger Technology (DLT), Cryptography, and Consensus Mechanisms. Types of Blockchains - Public, Private, Consortium, and Hybrid Block chains. Blockchain Structure - Blocks, Chains, Nodes, and Transactions. Cryptographic Foundations - Hash Functions, Digital Signatures, Public and Private Keys. Consensus Algorithms - Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS. Smart Contracts - Definition, Creation, Execution, and Security Issues. Overview of Major Blockchain Platforms - Bitcoin, Ethereum, Hyperledger.	
Unit 2: Blockchain and Cryptocurrencies	**15 Hrs**
Blockchain Networks - Nodes, Peer-to-Peer Networks, and Distributed Consensus. Security in Blockchain - Threats, Attacks, and Countermeasures. Blockchain Use Cases - Financial Services, Supply Chain, Healthcare. Introduction to Cryptocurrencies - Bitcoin and Altcoins. Bitcoin Architecture - Blockchain, Mining, Wallets, and Transactions. Ethereum and Smart Contracts - Solidity, DApps, and Gas. Cryptocurrency Wallets - Types, Security, and Key Management.	
Unit 3: Blockchain Development and Implementation	**15 Hrs**
Introduction to Blockchain Development - Tools, Platforms, and IDEs. Blockchain Development Languages - Solidity, Vyper, Go, and JavaScript. Building Smart Contracts - Basics, Writing, and Deploying.	

Developing DApps - Frontend, Backend, and Smart Contract Integration. Ethereum Development Environment - Truffle, Ganache, Remix. Hyperledger Fabric - Architecture, Components, and Development. Testing Blockchain Applications - Unit Tests, Integration Tests .	
Unit 4: Advanced Topics and Future Directions in Blockchain	**15 Hrs**
Blockchain in IoT - Use Cases, Challenges, and Solutions. Blockchain and Big Data - Integration, Analytics, and Use Cases. Blockchain in AI - Synergies, Applications, and Challenges. Blockchain and Cloud Computing - Decentralized Cloud Solutions. Green and Sustainable Blockchain Technologies. Quantum Computing and its Impact on Blockchain . Future Directions - Web 3.0, Decentralized Identity, and Tokenization of Assets.	

Textbooks:

1. "Blockchain Technology: Concepts and Applications" by Kumar Saurabh and Ashutosh Saxena, McGraw-Hill Education (2020).
2. "Cryptocurrency and Blockchain Technology" by Shaik Nasrullah and M. Balamurugan, Pearson (2021).
3. "Blockchain and Cryptocurrency" by B. B. Gupta and Hemraj Saini, PHI Learning (2020).

Reference Books:

1. "Cryptography and Blockchain Technology" by Atul Kahate, McGraw-Hill Education (2018).
2. "Blockchain: Principles and Applications" by Umesh Kumar Singh and Kavita Rani, Pearson (2020).
3. "Blockchain Technology and Applications" by M. S. Kiruthika and B. Prabu, PHI Learning (2021).

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand the fundamental concepts, cryptographic principles, types, and architectures of blockchain systems and analyze major blockchain platforms.
CLO2: Evaluate the structure, operations, and applications of cryptocurrencies and identify the security implications in blockchain networks.
CLO3: Apply blockchain programming languages and development frameworks to build and deploy smart contracts and decentralized applications (DApps).
CLO4: Apply blockchain programming languages and development frameworks to build and deploy smart contracts and decentralized applications (DApps).

LEVEL OF CLO-PLO MAPPING TABLE

Unit wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCADBC125.1	3	2	2	2	3	2	2	2	2	3	**2.3**
MMCADBC125.2	3	3	2	2	3	2	2	2	3	3	**2.5**
MMCADBC125.3	3	3	3	2	3	1	2	3	2	3	**2.5**
MMCADBC125.4	3	2	2	2	3	2	3	3	3	2	**2.5**

Average (PLO)	**3.0**	**2.5**	**2.25**	**2.0**	**3.0**	**1.75**	**2.25**	**2.5**	**2.5**	**2.75**	**2.45**

COURSE TITLE: Cyber Security and Digital Forensics

Course Code: MMCADCS125						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**	

Course Objectives

- To describe the fundamentals of cybersecurity
- To analyze security challenges faced by different IT components
- To describe the fundamentals of digital forensics
- To apply digital forensic methods to analyze disk drives and file systems
- To use computer forensic tools to perform file system forensics
- To describe anti-forensic techniques, types and tools

Course Content	**TEACHING HOURS**
UNIT 1: Cyber security	**15 Hrs**
Introduction – history, relevance, major incidents. Basic terminology. CIA triad. Cyber attacks and classification. Cyber crimes and classification. Cyber laws and penalties. IT Act, 2000. System security and Software security. Web security and Network security.	
UNIT 2: Digital forensics	**15 Hrs**
Introduction – principles, procedures, phases, types. Sources of digital evidence and chain-of-custody. Data acquisition and validation. Computer forensic tools (CFTs). Timeline analysis. Proactive and reactive forensics.	
UNIT 3: File system forensics	**15 Hrs**
Storage drive design and working. Volume analysis, PC-based partitions, Server-based partitions. File system analysis, FAT file system concepts, data structures and analysis. Using CFTs to perform forensic analysis of the FAT file system.	
UNIT 4: Anti-forensics	**15 Hrs**
Introduction, artifact-wiping, data-hiding, cryptography, steganography, trail obfuscation, attacking CFTs. Anti-forensics tools. Anti-forensics countermeasures. Forensic readiness.	

Textbooks:

1. E. Casey, Handbook of Digital Forensics and Investigation, Academic Press, 2010.

Reference Books:

1. B. Carrier, File System Forensic Analysis, Addison-Wesley, 2005.
2. J.R. Vacca and K. Rudolph, System Forensics, Investigation and Response, Jones and Bartlett Learning, 2011.
3. M. T. Britz, Computer Forensics and Cyber Crime, Pearson, 2013.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand fundamental concepts of cyber security, including attack classifications, cyber laws, and the principles of system, software, web, and network security.

CLO2: Apply digital forensic procedures and tools to identify, preserve, and analyze digital evidence while maintaining legal and ethical standards.

CLO3: Perform forensic analysis of file systems, especially FAT, by understanding storage structures, volume formats, and using appropriate forensic tools.

CLO4: Analyze anti-forensics techniques and apply countermeasures to improve forensic readiness and maintain evidence integrity in digital investigations.

CLO-PLO Matrix for the Course

	PLOs										
Unit-Wise CLOs	1	2	3	4	5	6	7	8	9	10	**Avg (CLO)**
MMCADCS125.1	3	2	2	1	3	3	2	2	1	3	**2.2**
MMCADCS125.2	3	3	3	1	3	3	2	2	2	3	**2.5**
MMCADCS125.3	3	3	3	1	3	2	2	2	2	3	**2.4**
MMCADCS125.4	3	2	3	1	3	3	2	3	2	3	**2.5**
Avg (PLO)	**3.0**	**2.5**	**2.8**	**1.0**	**3.0**	**2.75**	**2.0**	**2.25**	**1.75**	**3.0**	**2.4**

COURSE TITLE: Research Methodology

Course Code: MMCACRM125	**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 30	**External**	**36**	**-**
	Internal	**14**	**-**

Lecture(L):	**2**	**Practicals(P):**	**Tutorial(T):**	**-**	**Total Credits**	**2**	

Course Objectives

- To develop knowledge about selecting and defining research problems, and approaches to problem-solving in research.
- To learn to conduct effective literature reviews, handle data responsibly, and practice ethical research.
- To understand the basics of patents, copyrights, and trademarks, and their significance in innovation.
- To IP laws, and technology transfer for real-world applications.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Research Methodology	**15 Hrs**
Meaning of research, objectives and motivation of research: Types of research: fundamental, applied, descriptive, analytical; Research process and formulation of research problem: Criteria for good research	
UNIT 2: Literature Review and Technical Writing	**15 Hrs**
Searching for literature: digital libraries, journals, databases: Literature survey and review techniques: Technical writing: structure of a research paper, proposal, thesis, and report writing. Journal metrics, indexing, and their significance in defining the quality of a journal.	

Textbooks

1. C.R. Kothari, *Research Methodology: Methods and Techniques*, New Age International.
2. Ranjit Kumar, *Research Methodology: A Step-by-Step Guide for Beginners*, Sage Publications.

ReferenceBooks

1. Wayne Goddard & Stuart Melville, *Research Methodology: An Introduction.*.
2. T.N. Huckin and L.A. Olsen, *Technical Writing and Professional Communication*.

COURSE LEARNING OUTCOMES(CLO):

CLO1: Understand the principles of research methodology, including problem identification, ethical considerations, and literature review techniques.

CLO2: Apply appropriate data collection methods, statistical analysis techniques, and digital tools to produce well-structured and ethical research reports.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	**PLOs**

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCACRM125.1	3	2	2	2	3	3	3	2	3	2	**2.5**
MMCACRM125.2	3	3	3	3	3	2	3	2	3	2	**2.7**
Avg (PLO)	**3.0**	**2.5**	**2.5**	**2.5**	**3.0**	**2.5**	**3**	**2**	**3**	**2**	**2.6**

COURSE TITLE: JAVA PROGRAMMING (Lab)

Course Code: MMCALJP125		**Examination Scheme**	
Total number of Lab Hours: 60 hours		**External**	**36**
		Internal	**14**

Lecture (L):	**-**	**Practicals (P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**2**

Course Objectives:

- To understand how to download and install the latest JDK version (preferably JDK 8 or above) and configure the PATH variable for Java binaries.
- To develop proficiency in writing, compiling, and executing simple Java programs, including basic I/O operations, arithmetic calculations, and conditional statements.
- To gain familiarity with object-oriented programming by defining classes, using concepts like encapsulation, constructors, method overloading, and beginning to apply exception handling in Java programs.
- To learn how to utilize networking classes in Java to establish communication between applications, including sending and receiving text messages over a network connection.

- **Week 1:**
  - Download latest version of Java Development Kit (JDK), preferably JDK8 or above (Please visit https://java.com/en/download/).
  - Follow the instructions that appear during the Installation ofJDK8, and set PATH variable to the appropriate directory location as instructed in the lecture.
- **Week 2:**
  - Write a Java program that displays “hello world!” on the screen.
  - Write a Java program that receives two integer numbers via keyboard, does their summation, and displays the result. Ensure that only integer values are processed.
  - Write a Java program that prints the season name corresponding to its month number using Ifelse and switch-case statements.
  - Write a Java program that sorts (using bubble sort) an integer array using for loop.
  - Write a Java program that calculates factorial of a number (inputted via keyboard) recursively.
  - Write a Java program that creates a 2D integer array with 5 rows and varying number of columns in each row. Using ‘for each’ variant of for loop display each element of every row.
- **Week 3:**
  - Write a Java program that creates a Class, namely Student.
    - Ensure that Age instance variable of the Class is never accessed directly, and its value is never less than 4 and greater than 40 for any Object of the Class (use methods to validate and assign the value).
    - Ensure that the constructor always assigns a unique value to Enrollment_No instance variable for every Object of the Class (use a static class variable for counting objects, say Object_Counter).
    - Ensure that when an Object is removed, the Object_Counter is automatically decremented (use finalize()), and whenever required the variable can only be accessed using a method even without an Object reference (make the counter private and use a static method to access it).
  - Write a Java program in which a Class overloads a method sum(), which takes 2 parameters. The overloaded methods should perform summation of either integer or

floating-point values

- **Week 4:**
    - Write a Java program that creates a Class namely A that has a private instance variable and method, a protected instance variable and method, a default instance variable and method, and a public instance variable and method. Create another Class say B that inherits from A.
        - Show that all except private members are inherited.
        - Show that an inherited instance variable can be shadowed (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
        - Show that an inherited method can be overridden (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
        - Show that the reference variable of type A or B can't access an overridden method of A in the Object of B.
        - Show that the reference variable of type A can access a shadowed data member of A in the Object of B.
- **Week 5:**
    - Write a Java program that creates a Class in which a method asks the user to input 2 integer values, and calls another member function (say div()) to divide the first inputted number by the second number (by passing them as parameters). Handle an exception that can be raised in div() when the denominator equals zero (use try-catch statement).
    - Modify the above Java program so that it also creates a Custom Exception that is thrown by div() when the denominator value is 1 (use throw). Handle the exception. c.
    - Modify the above Java program so that the exception-handling in not performed by div() rather it only species all the possible exceptions it may throw (use throws). And, the method that calls div() does the exception handling.
- **Week 6:**
    - Create a Java Package (say pack1) that contains 3 Classes (say A, B and C). Write a Java program that uses this package after setting the CLASSPATH variable. Following scenarios must be considered individually:
        - Importing the whole package (all the 3 classes)
        - Importing only specific class (say Class A only)
    - Create another Package (say pack2) that contains same number of classes, and same definition for each class, as that of pack1. Write a Java program that imports all classes from both pack1 and pack2 while ensuring that the name conflicts are not encountered while accessing any of these classes.
- **Week 7:**
    - Write a Java program to count the number of words in a string that is passed as a command line argument.
    - Write a Java program to check whether a string is palindrome or not.
    - Write a Java program to count the total number of occurrences of a given character in a string.
    - Write a Java program to convert a string to char array.
- **Week 8:**
    - Write a Java program that creates a Class that extends a Thread class. Create 3 objects of the class, each starting a new thread and each thread displaying "I am Thread: " in an infinite loop. The displayed text must be suffixed by the unique name of the thread.

  - Write a Java program that creates a Class that implements interface Runnable, and does the same as the above program.
  - Write a Java program to implement a solution for producer-consumer problem using synchronization and inter-process communication in Threads.

- **Week 9:**
  - Write a Java program to open and read a file (filename is passed as command line argument), and displays the number of words in the file?
  - Write a Java program to copy a file. The source and destination filenames are passed as command line arguments.
- **Week 10:**
  - Java program to determine number of bytes written to file using DataOutputStream
  - Java program to read text from file from a specified index or skipping byte using FileInputStream
- **Week 11:**
  - Create a Java AWT program to handle a button click event using ActionListener.
  - Write a program to display a message when the mouse is clicked anywhere on the frame using MouseListener.
  - Develop a program to detect and display which key is pressed using KeyListener.
- **Week 12:**
  - Create a GUI application where clicking a button increases a counter displayed on the screen.
  - Write a Java program to change the background color of a frame when a button is clicked.
  - Build an application where hovering the mouse over a button displays a tooltip using mouse events.
  - Design a login form using AWT, and validate input fields using event handling.
- **Week 13:**
  - Write a Java program (client) that sends a text message to another Java program (server), which receives and displays it.
  - Modify the above Java programs so that each of the two programs is able to send and receive the text messages.
- **Week 14:**
  - Write a Java program (a client) that opens a connection to https://www.Internic.net website and displays information about www.google.com.
  - Write a Java program (Client) that sends a text message to another Java program (Server), and the Server displays an acknowledgement message on receiving it.
  - Write a Java program (Client) that sends a text string to another Java program (Server), which receives it and sends back the reverse string of the received string.

Note: The Lab course shall be conducted over a course of 14 weeks, with a minimum of 2 labs per week.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Set up the Java development environment and apply fundamental programming constructs such as variables, loops, conditionals, arrays, and methods.

CLO2: Implement object-oriented programming concepts including inheritance, encapsulation, method overloading/overriding, exception handling, and packages in Java.

CLO3: Manipulate strings, threads, and file input/output operations to develop multi-threaded and file-handling Java applications.

CLO4: Design GUI-based and networked Java applications using event handling, AWT, and socket programming.

CLOs	**PLO1**	**PLO2**	**PLO3**	**PLO4**	**PLO5**	**PLO6**	**PLO7**	**PLO8**	**PLO9**	**PLO10**	**Average (CLO)**
MMCALJP125.1	3	3	2	1	2	0	1	1	1	1	**1.5**
MMCALJP125.2	3	3	3	1	3	1	1	2	2	1	**2**
MMCALJP125.3	2	3	3	1	3	0	1	2	2	1	**1.8**
MMCALJP125.4	2	3	3	2	3	1	2	3	3	3	**2.5**
Average(PLO)	**2.5**	**3**	**2.75**	**1.25**	**2.75**	**0.5**	**1.25**	**2**	**2**	**1.5**	**1.95**

COURSE TITLE: Machine Learning Lab

Course Code: MMCALML125						**Examination Scheme**	**T**	**P**
Total number of Practical Hours: 60						**External**	**0**	**36**
						Internal	**0**	**14**
Lecture (L):	**0**	**Practical (P):**	**2**	**Tutorial (T):**	**0**	**Total Credits**		**2**

Course Learning Objectives:

- To introduce students to Python-based machine learning tools and environments.
- To provide hands-on experience in data preprocessing, visualization, and model building.
- To implement supervised and unsupervised machine learning techniques using real-world datasets.
- To evaluate machine learning models using appropriate metrics and improve model performance.

Practical’s

Week 1: Python and ML Tools Setup

- Install a Python distribution suitable for Machine Learning tasks.
- Explore and demonstrate basic functions of NumPy, Pandas, Matplotlib, scikit-learn, and SciPy.

Week 2: Google Colab

- Demonstrate the use of Google Colab and explain its benefits for Machine Learning development.
- Create and perform basic operations in a Colab Notebook, including code execution and file sharing.

Week 3: Data Handling and Visualization

- Write a Python program to import and export data using Pandas.
- Write a Python program to demonstrate various data visualization techniques using Matplotlib/Seaborn.

Week 4: Data Preprocessing

- Demonstrate various data preprocessing techniques (handling missing data, normalization, etc.) on a given dataset.
- Apply data preprocessing methods to the IRIS dataset using scikit-learn.

Week 5: Data Analysis

- Plot 2D views of the IRIS dataset using Matplotlib.
- Download and scan a dataset (e.g., IRIS), list features and types, analyze distributions, and identify outliers.

Week 6: Classification with Decision Tree and KNN

- Implement the decision tree using the ID3 algorithm.
- Implement the K-Nearest Neighbour algorithm for the IRIS dataset classification task.

Week 7: Exploring KNN Parameters

- Analyze the effect of various parameters on KNN algorithm performance.
- Compare the effect of different distance measures (Manhattan, Euclidean, etc.) on KNN classification.

Week 8: Regression Techniques

- Implement linear regression on a given dataset.
- Implement logistic regression on a given dataset.

Week 9: Model Evaluation

- Compute confusion matrix and evaluate performance (TP, FP, TN, FN, Accuracy, Precision, Recall, Error Rate) using logistic regression results.

Week 10: Clustering with K-Means

- Apply K-Means clustering on the IRIS dataset and analyze results.
- Evaluate the effect of changing K-Means parameters like number of clusters and initialization.

Week 11: Advanced Classification Tasks

- Build and train a Support Vector Machine (SVM) for a classification task.
- Build a classification model to predict loan approval using real-world data.

Week 12: Dimensionality Reduction using PCA

- Implement PCA on the IRIS dataset and visualize the principal components.
- Plot and interpret the first two principal components. Explain their significance in dimensionality reduction.

Week 13: Naïve Bayes Algorithm

Implement Naïve Bayes from scratch and answer the following:

- How does the algorithm work?
- What are the variations of Naïve Bayes?
- What are the advantages and limitations?
- What are the steps to implement it?
- How can it be improved?
- When should it be used?

Week 14: Model Evaluation and Real-World Applications

- Evaluate the Naïve Bayes model using confusion matrix and performance metrics.
- Implement anomaly detection on a dataset using Python.
- Solve a real-world problem using three different ML techniques: Logistic Regression, Support Vector Machines, and K-Means Clustering.

Textbooks

1. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2nd Edition.
2. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education.

Reference Books

1. Andreas Müller and Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press.
3. Online Dataset Repositories: UCI Machine Learning Repository

COURSE LEARNING OUTCOMES (CO):
CLO1: Set up a machine learning environment using Python and Google Colab.
CLO2: Perform data preprocessing, transformation, and visualization using appropriate libraries.
CLO3: Apply various supervised and unsupervised machine learning algorithms**.**
CLO4: Evaluate classification and clustering models using performance metrics.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average (CLO)**
MMCALML125.1	3	2	2	2	2	1	2	2	2	2	**2.0**
MMCALML125.2	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCALML125.3	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCALML125.4	3	3	3	2	3	1	2	2	2	3	**2.4**
Average (PLO)	**3.0**	**2.75**	**2.75**	**2.0**	**2.75**	**1.0**	**2.0**	**2.0**	**2.0**	**2.75**	**2.3**

SEMESTER-II

COURSE TITLE: Design and Analysis of Algorithm

Course Code: MMCACDA225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To gain a solid foundation in algorithms, their analysis, and the growth of functions.
- To apply asymptotic notations and techniques to study the time and space complexity of algorithms.
- To explore and apply methods such as recurrences, the Master Method, and randomized algorithms.
- To utilize divide and conquer, greedy, dynamic programming, backtracking, and branch and bound strategies to solve complex problems.
- To learn about P, NP, NP-hard, and NP-complete problems, and understand the significance of Cook’s Theorem.
- To evaluate the need for and implement approximation algorithms for solving complex optimization problems.

Course Content	TEACHING HOURS
UNIT I: Fundamentals of Algorithm Analysis	**15 Hrs**
Introduction to Algorithms, Analysis of Algorithms, Growth of Functions, Asymptotic notations. Recurrences, Substitution method, Iteration method, Recursion trees, The Master Method, Time and Space Complexity study of some basic algorithms.	
UNIT II: Advanced Algorithmic Techniques	**15 Hrs**
Randomized Algorithms: Identifying the repeated element, Primality testing, Advantages and Disadvantages. Divide and Conquer Strategy: Binary search, Quick sort, Merge sort, Greedy Method, General method, Knapsack problem, Single source shortest paths.	
UNIT III: Optimization and Search Strategies	**15 Hrs**
Dynamic programming Strategy: All pair shortest paths, Traveling salesman problems. Backtracking Strategy: 8-Queen problem, Sum of subsets, Knapsack problem. Branch and Bound Strategy: Least Cost Branch and Bound, 8-Queen Problem	
UNIT IV: Computational Complexity and Approximation Algorithms	**15 Hrs**
Lower boundary theory, Lower bound theory through reductions, P and NP problems. NP hard and NP complete problems, Cook’s Theorem, Approximate Algorithms and their need, The vertex Cover Problem, The traveling salesman problem, The subset sum problem	

Textbooks:

1. Horowitz, Sahni, Rajasekaran “Fundamentals of Computer Algorithms”,Galgotia Publications

Reference Books:

1. Coremen, Leiserson, Rivest,Stein, “Introduction to Algorithms”, 2nd edition, PHI.
2. Michael T. Goodrich, Roberto Tamassia “Algorithm Design and Applications” , Wiley
3. Aho, Hopcroft and Ullman, “The Design and Analysis of Computer Algorithms”, Pearson

COURSE LEARNING OUTCOMES (CLO):

CLO1: Students will be able to analyze the time and space complexity of algorithms using asymptotic notations and recurrence-solving techniques such as the substitution method, recursion trees, and the Master Method.

CLO2: Students will apply algorithmic paradigms like divide-and-conquer, greedy method, and randomization to solve computational problems such as sorting, shortest paths, and primality testing.

CLO3: Students will implement and evaluate optimization techniques using dynamic programming, backtracking, and branch-and-bound strategies for problems like TSP, 8-Queen, and knapsack.

CLO4: Students will classify computational problems based on complexity classes (P, NP, NP-Complete, NP-Hard) and apply approximation algorithms to solve intractable problems such as vertex cover and subset sum.

CLO-PLO Matrix for the Course

	PLO										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCACDA225.1	3	2	2	0	3	0	1	1	1	3	**1.6**
MMCACDA225.2	3	3	3	0	3	0	1	2	2	3	**2.0**
MMCACDA225.3	2	3	3	0	3	0	1	2	2	3	**1.9**
MMCACDA225.4	3	2	2	1	3	1	1	1	2	3	**1.9**
Avg (PLO)	**2.8**	**2.5**	**2.5**	**0.25**	**3.0**	**0.25**	**1.0**	**1.5**	**1.75**	**3.0**	**1.85**

COURSE TITLE: Mobile Application Development

Course Code: MMCACMA225						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practical (P):**	**2**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Objectives:

- To identify various concepts of mobile programming that make it unique from programming other platforms.
- To interpret features of Android Operating System.
- To critique mobile applications on their design pros and cons.
- To utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces.
- To program mobile applications for the Android operating system that use basic and advanced phone features.

COURSE CONTENT	TEACHING HOURS
UNIT 1: Mobile Application Development	**15 Hrs.**
Mobile Applications and Device Platforms, Alternatives for Building Mobile Apps, Comparing Native vs. Hybrid Applications, The Mobile Application Development Lifecycle, The Mobile Application Front-End and Back-End, Key Mobile Application Services, Introduction to Java, Java Setup and Program structure, Inheritance and Exception, Handling Events, Debugging, Introduction to Classes.	
UNIT 2: Introduction to Android	**15 Hrs.**
The Android Platform, Android SDK, Eclipse Installation, Android Installation, Building you First Android application, Understanding Anatomy of Android Application, Need of Android, Advanced Android Features, Tools and Software required for developing an Android application.	
UNIT 3: Android terminologies	**15 Hrs.**
Android terminologies, Application Context, Activities, Services, Intents, Android Storing and Retrieving data, Receiving and Broadcasting Intents, Content Provider, Android Manifest File and its common settings, Using Intent Filter, Permissions, Android Networking and Web.	
UNIT 4: Android User Interface Design Essentials	**15 Hrs.**
Android User Interface Design Essentials: Fundamental UI design, User Interface Screen elements, Designing User Interfaces with Layouts, Text View, List View, Grid View, Image View, Scroll View, Drawing and Working with Animation, SQLite Database, Creating and Connection of the database.	

Textbooks

1. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011)
2. Jerome DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition.

Reference Books

1. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd
2. Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
3. Hortan, John, “Android Programming for Beginners”, Packet Publication, 2015, ISBN: 978-1-78588-326-2.

COURSE LEARNING OUTCOMES (CO):

CLO1: Understand mobile app development fundamentals, Java basics, and development lifecycle.

CLO2: Set up Android development environment and build basic Android applications.

CLO3: Explain core Android components, data handling, intents, and permissions.

CLO4: Design Android user interfaces and implement database connectivity with SQLite.

LEVEL OF CO-PO MAPPING TABLE

	PLO										
UNIT-WISE CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCACMA225.1	3	2	2	0	0	0	0	0	0	0	**2.33**
MMCACMA225.2	0	0	3	3	3	0	0	0	0	0	**3.0**
MMCACMA225.3	0	0	0	0	3	3	2	0	0	0	**2.67**
MMCACMA225.4	0	0	0	0	0	0	0	3	2	0	**2.5**
Avg (PLO)	**1.5**	**1.0**	**1.67**	**1.5**	**2.0**	**1.5**	**1.0**	**1.5**	**1.0**	**0**	**2.63**

DCE-III

COURSE TITLE: Advanced Operating Systems

						Examination Scheme	T	P
Course Code: MMCADA0225								
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- To understand Fundamental Concepts of Operating Systems
- To develop Skills in Process Management and Synchronization
- To explore Distributed Operating Systems
- To gain Expertise in Deadlocks Management
- To explore Real Time Operating System

Course Content	TEACHING HOURS
UNIT 1: Introduction and Scheduling	**15 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.	
UNIT 2: Inter-Process Communication and Synchronization	**15 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.	
UNIT 3: Memory Management	**15 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.	
UNIT 4: Deadlocks	**15 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.	

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 LEARNING OUTCOMES (CLO):

CLO1: Identify OS types; apply scheduling and fault-tolerance in basic, distributed, and Real Time Operating System.

CLO2: Implement IPC and synchronization in centralized, distributed, and Real Time Operating System.

CLO3: Analyze memory management in traditional, distributed, and Real Time Operating System.

CLO4: Apply deadlock handling in centralized, distributed, and Real Time Operating System.

LEVEL OF CO-PO MAPPING TABLE

	PLO										
UNIT-WISE CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADAO225.1	3	2	2	0	0	0	0	0	0	0	**2.33**
MMCADAO225.2	0	0	3	3	2	0	0	0	0	0	**2.67**
MMCADAO225.3	0	0	0	0	3	2	0	0	0	0	**2.50**
MMCADAO225.4	0	0	0	0	0	2	3	3	0	0	**2.67**
Avg (PLO)	**1.5**	**1.0**	**1.67**	**1.5**	**2.5**	**2.0**	**1.5**	**1.5**	**0**	**0**	**2.54**

COURSE TITLE: Digital Image Processing

Course Code: MMCADDI225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60 **Total number of Practical Hours: -**						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- To develop a thorough understanding of the fundamental concepts and theories in image processing, including pixel representation, color spaces, and digital image formation
- To equip students with the technical skills to apply various image processing techniques such as image transformations, filtering, enhancement, and segmentation using appropriate software tools.
- To 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.
- To 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.	**15 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	**15 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.	**15 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.	**15 Hrs.**
Edge detection: Basic Formulation: Detecting Points and Lines, Edge Models; Gradient andits 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 LEARNING OUTCOMES (CLO):

CLO1: Understanding the fundamental concepts of digital image processing, including image formation, digitization, and pixel relationships.

CLO2: Apply spatial and frequency domain techniques for image enhancement using filtering and transformation methods.

CLO3: Analyze image degradation models and perform restoration and morphological operations for noise removal and structure preservation.

CLO4: Implement edge detection and image segmentation techniques for identifying and extracting regions of interest in digital images.

LEVEL OF CLO-PLO MAPPING TABLE

CLOs	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	Average (CLO)
MMCADDI225.1	3	1	2	0	2	0	1	1	0	2	1.2
MMCADDI225.2	2	3	2	1	3	0	1	2	1	3	1.8
MMCADDI225.3	3	3	3	1	3	1	2	2	1	3	2.2
MMCADDI225.4	2	3	3	1	3	1	2	3	2	3	2.3
Average(PLO)	2.5	2.5	2.5	0.75	2.75	0.5	1.5	2	1	2.75	1.8

COURSE TITLE: Decision Support Systems

						Examination Scheme	T	P
Course Code: MMCADDS225								
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- To gain a comprehensive understanding of Decision Support Systems, including their importance in enhancing decision-making processes within organizations.
- To 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.
- To 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.
- To 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.
- To 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.

Course Content	TEACHING HOURS
UNIT 1: Decision Making	**15 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	**15 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	**15 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	**15 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 LEARNING OUTCOMES (CLO):

CLO1: Understand decision support systems (DSS), their development methodologies, and technology platforms to aid in effective decision-making.

CLO2: Apply modeling techniques in DSS to address scenarios involving certainty, uncertainty, and risk.

CLO3: Design and implement DSS solutions using appropriate tools, platforms, and methodologies.

CLO4: Analyze enterprise-level DSS and knowledge management systems for enhanced organizational decision-making and collaboration.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCADDS225.1	3	2	2	2	2	1	2	2	2	2	**2.0**
MMCADDS225.2	3	3	2	2	3	1	2	2	2	3	**2.3**
MMCADDS225.3	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCADDS225.4	3	2	3	3	3	2	3	2	2	2	**2.5**
Average (PLO)	**3.0**	**2.5**	**2.5**	**2.25**	**2.75**	**1.25**	2.25	2	2	2.5	2.3

COURSE TITLE: Cryptography and Network Security

Course Code: MMCADCN225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
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 I: Basics of Security and Classical Encryption	**15 Hrs.**
Introduction to Information and Network Security, Security Goals: Confidentiality, Integrity, Availability, Types of Attacks and Threats, Basics of Number Theory for Cryptography, Classical Encryption Techniques: Substitution, Transposition, One-Time Pad	
UNIT II: Modern Cryptography	**15 Hrs.**
Symmetric Encryption: DES, AES, and Modes of Operation, Stream Ciphers and Pseudorandom Number Generators, Asymmetric Encryption: RSA, Diffie-Hellman, ElGamal, and ECC	
UNIT III: Data Integrity and Digital Signatures	**15 Hrs.**
Cryptographic Hash Functions (SHA-1, SHA-3), Message Authentication Codes (HMAC, CMAC), Digital Signatures: RSA, ElGamal, ECDSA, Key Management Basics	
UNIT IV: Network Security Practices	**15 Hrs.**
Secure Communication: HTTPS, TLS, SSH, Email and IP Security, Firewalls and 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 LEARNING OUTCOMES (CO):

CLO1: Students will be able to explain fundamental concepts of information and network security, including security goals, classical encryption methods, and number theory used in cryptography.

CLO2: Students will apply symmetric and asymmetric cryptographic algorithms such as AES, RSA, and ECC to ensure secure communication.

CLO3: Students will analyze and implement data integrity techniques using hash functions, MACs, and digital signature schemes.

CLO4: Students will describe and evaluate network security practices including TLS, SSH, firewalls, and intrusion detection systems.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	**PLO 1**	**PLO 2**	**PLO 3**	**PLO 4**	**PLO 5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**	**PLO 10**	**Avg (CLO)**
MMCADCN225.1	3	2	1	1	2	2	1	1	1	2	**1.6**
MMCADCN225.2	3	3	2	1	3	2	1	2	2	3	**2.2**
MMCADCN225.3	3	3	2	1	3	3	2	2	2	3	**2.4**
MMCADCN225.4	2	3	2	2	2	2	1	1	1	2	**1.8**
Average (PLO)	**2.75**	**2.75**	**1.75**	**1.25**	**2.5**	**2.25**	**1.25**	**1.5**	**1.5**	**2.5**	**2**

DCE-IV

<table>
<tr><td colspan="9"><b>COURSE TITLE: Advanced Computer Networks</b></td></tr>
<tr><td colspan="6"><b>Course Code: MMCADAC225</b></td><td><b>Examination Scheme</b></td><td><b>T</b></td><td><b>P</b></td></tr>
<tr><td colspan="6" rowspan="2"><b>Total number of Lecture Hours:60</b></td><td><b>External</b></td><td><b>72</b></td><td><b>-</b></td></tr>
<tr><td><b>Internal</b></td><td><b>28</b></td><td><b>-</b></td></tr>
<tr><td><b>Lecture (L):</b></td><td><b>4</b></td><td><b>Practical’s (P):</b></td><td><b>-</b></td><td><b>Tutorial (T):</b></td><td><b>-</b></td><td><b>Total Credits</b></td><td colspan="2"><b>4</b></td></tr>
</table>

Course Objectives

- To gain a comprehensive understanding of the core principles of computer networking, including protocol design, protocol layering, algorithm design, and performance evaluation.
- To acquire detailed knowledge of the OSI model and TCP/IP protocol suite and understand the design issues and protocols used in the data link layer and MAC sublayer.
- To Understand the design issues of the network layer, including various routing algorithms and congestion control mechanisms.
- To learn about the protocols used in the transport and application layers, including their design and functionality.

Course Content	**TEACHING HOURS**
UNIT I:	**-15 Hrs**
Introduction: Components of Network, Topologies, Categories of Networking: LAN, WAN, MAN. Uses of Networks. Reference Models: TCP/IP Model, The OSI Model, and Comparison of the OSI and TCP/IP reference model. Architecture of Internet. **Transmission Media:** Guided transmission media, Wireless transmission media, Radio Transmission, Microware Transmission, Infrared Transmission and Light Transmission, Digital Modulation and Multiplexing, Switching.	
UNIT II:	**-15 Hrs**
Data Link Layer: Design issues, Error Detection & Correction, Elementary Data Link Layer Protocols, Sliding window protocols and SONET **Medium Access Control Sub layer:** The Channel Allocation problem and Multiple Access Protocols, Ethernet. Multiple Access Protocols - ALOHA, CSMA,CSMA/CD, CSMA/CA, Collision free protocols, Ethernet- Physical Layer, Ethernet Mac Sub layer, Data link layer switching: Use of bridges, learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways.	
UNIT III:	**-15 Hrs**
Network Layer: Network Layer Design issues, store and forward packet switching connection less and connection oriented networks-routing algorithms-optimality principle, shortest path, flooding, Distance Vector Routing, Link State Routing, Path Vector Routing, Hierarchical Routing; Congestion control algorithms, IP addresses, CIDR, Subnetting, SuperNetting, IPv4, Packet Fragmentation, IPv6 Protocol, Transition from IPv4 to IPv6, ARP, RARP, OSPF , BGP and Traffic Prioritization.	
UNIT IV:	**-15 Hrs**
Transport Layer: Services provided to the upper layer’s elements of transport protocol addressing connection establishment, Connection release, Error Control & Flow Control, Crash Recovery. The Internet Transport Protocols: UDP, Introduction to TCP, The TCP Service Model, The TCP Segment Header, The Connection Establishment, The TCP Connection Release, The TCP Sliding Window, The TCP Congestion Control Algorithm, Socket Programming. **Application Layer**: Introduction, providing services, Applications layer paradigms: Client server model, HTTP, E-mail, WWW, TELNET, DNS.	

Textbooks

1. Kurose, James F., and Keith W. Ross. *Computer Networking: A Top-Down Approach.* 8th ed., Pearson, 2021.
2. Stallings, William. *Data and Computer Communications.* 11th ed., Pearson, 2022.
3. Tanenbaum, Andrew S., and David J. Wetherall. *Computer Networks.* 5th ed., Pearson, 2013.

Reference Books

1. Forouzan, Behrouz A. *TCP/IP Protocol Suite.* 5th ed., McGraw-Hill Education, 2023.
2. Comer, Douglas E. *Internetworking with TCP/IP: Principles, Protocols, and Architecture.* Vol. 1, 6th ed., Pearson, 2021.
3. Stallings, William. *Wireless Communications and Networks.* 2nd ed., Pearson, 2005.

COURSE LEARNING OUTCOMES (CLO):

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

CLO1: List the functionalities of different layers in both the OSI and TCP/IP reference models.
CLO2: Identify data link layer design issues and apply error detection and correction techniques.
CLO3: Describe the principles of switching and routing algorithms used in computer networks.
CLO4: Distinguish between TCP and UDP formats and procedures, understanding their respective uses and characteristics.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	PLO 10	Avg (CLO)
MMCADAC225.1	3	1	1	1	1	1	1	1	1	1	**1.2**
MMCADAC225.2	2	3	3	1	2	2	2	1	1	3	**2**
MMCADAC225.3	3	3	3	3	2	1	2	2	2	3	**2.4**
MMCADAC225.4	3	2	2	2	2	1	1	1	1	2	**1.7**
Average (PLO)	**2.75**	**2.25**	**2.25**	**1.75**	**1.75**	**1.25**	**1.5**	**1.25**	**1.25**	**2.25**	**1.38**

COURSE TITLE: Cloud Computing

Course Code: MMCADCC225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60 **Total number of Practical Hours: -**						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

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

Course Content	TEACHING HOURS
UNIT 1: CLOUD COMPUTING FUNDAMENTALS	**15 Hrs.**
Basic Concepts and Terminology, Goals and benefits, Risks and Challenges, Roles and boundaries, Cloud characteristics. Cloud Delivery Models: IaaS, PaaS, Saas. Cloud Deployment Models: Public, private and Hybrid Cloud.	
UNIT 2: CLOUD-ENABLING TECHNOLOGIES & SECURITY FOUNDATIONS	**15 Hrs.**
Virtualization fundamentals: Hypervisors, VM provisioning, isolation Web technologies and multitenancy Service-oriented architecture (SOA) in cloud Network, storage, and broadband infrastructure Cloud security basics, Threat Agents ,Cloud Security Threats	
UNIT 3: CLOUD COMPUTING MECHANISMS	**15 Hrs.**
Cloud Infrastructure Mechanism: Virtual Server, Cloud Storage Device, Cloud Usage Monitor, Resource Replication. Specialized Cloud Mechanisms: Load balancing, failover, replication, caching, SLA monitoring, billing, auto-scaling Cloud Management Mechanisms: SLA Management System and Billing Management System	
UNIT 4: CLOUD COMPUTING ARCHITECTURE	**15 Hrs.**
Fundamental Architectures: Workload Distribution Architecture, Resource Pooling Architecture, Dynamic Scalability Architecture, Service Load Balancing Architecture, Cloud Bursting Architecture. Advanced Architectures: Hypervisor Clustering ,Load Balanced Virtual Server Instances Architecture, Non-Disruptive Service Relocation Architecture, Zero Downtime Architecture.	

Textbooks

1. Gautam Shroff, “Enterprise Cloud Computing: Technology, Architecture, Applications”, Cambridge University Press; 2nd Edition [ISBN: 9780521137355], 2023.
2. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach” McGraw-Hill Education; 2nd Edition [ISBN: 9780071826400], 2018.

3. Dimitris N. Chorafas, “Cloud Computing Strategies” CRC Press; 2nd Edition [ISBN: 9780367338611], 2021.

Reference Books

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

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understanding of the fundamental concepts of cloud computing, including cloud models (IaaS, PaaS, SaaS), deployment types, and their impact on business agility, performance, and security.

CLO2: Analyse the role of virtualization in cloud computing and evaluate the deployment and functionality of cloud-based web services.

CLO3: Assess the reliability, scalability, and economic aspects of cloud service management using contemporary tools and platforms.

CLO4: Design, develop, and deploy applications using modern cloud platforms such as AWS, Azure, and Google App Engine.

LEVEL OF CLO-PLO MAPPING TABLE

	PLOs										
CLOs	**PLO1**	**PLO2**	**PLO3**	**PLO4**	**PLO5**	**PLO6**	**PLO7**	**PLO8**	**PLO9**	**PLO10**	**Average (CLO)**
MMCADCC225.1	3	1	2	0	2	1	1	1	0	2	**1.3**
MMCADCC225.2	2	3	2	1	2	0	1	1	1	3	**1.6**
MMCADCC225.3	2	2	3	1	3	2	2	2	1	3	**2.1**
MMCADCC225.4	2	3	3	1	3	1	2	3	2	3	**2.3**
Average(PLO)	**2.25**	**2.25**	**2.5**	**0.75**	**2.5**	**1**	**1.5**	**1.75**	**1**	**2.75**	**1.8**

COURSE TITLE: Linux Programming

Course Code: MMCADLP225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

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

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Linux	**15 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	**15 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	**15 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	**15 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.

Reference Books:

1. S. Veeraraghavan. Shell Programming in 24 hours. SAMS/Techmedia, 2007.
2. R. Love. Linux Kernel Development. Addison-Wesley, 2010.

To be effective from year-2025

3. J. Blanchette, M. Summerfield. C++ GUI Programming with Qt3. Prentice Hall, 2004.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand the structure, utilities, and administrative functionalities of the Linux operating system, including user and process management.

CLO2: Apply shell scripting constructs to automate tasks and develop basic command-line applications using loops, conditionals, and functions.

CLO3: Demonstrate understanding of Linux kernel architecture and perform kernel extension through system calls and modules.

CLO4: Design and implement basic graphical user interface (GUI) applications using the toolkit and signal-slot mechanisms.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCADLP225.1	3	3	3	1	3	2	2	2	1	3	**2.3**
MMCADLP225.2	3	3	3	1	3	2	2	3	1	3	**2.4**
MMCADLP225.3	3	3	3	1	3	2	2	3	2	3	**2.5**
MMCADLP225.4	2	3	3	2	3	1	2	3	1	3	**2.3**
Average (PLO)	**2.75**	**3.0**	**3.0**	**1.25**	**3.0**	**1.75**	**2.0**	**2.75**	**1.25**	**3.0**	**2.38**

COURSE TITLE: Theory of Computation

Course Code: MMCADTC225						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practical (P):**	**0**	**Tutorial (T):**	**0**	**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, undesirability, reduction techniques, and complexity theory foundations.

COURSE CONTENT	TEACHING HOURS
UNIT 1: Introduction to Computation	**15 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	**15 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	**15 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	**15 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.	

Textbooks

1. New York: Wiley. Linz, Peter. An introduction to formal languages and automata. Jones & Bartlett Learning. Seventh Edition.

2. "Introduction to the Theory of Computation" by Michael Sipser, Third Edition.

Reference Books

1. Cohen, Daniel IA, Introduction to computer theory, 2nd Edition.
2. Parkes, Alan P. Introduction to languages, machines and logic: computable languages, abstract machines and formal logic. Springer Science & Business Media, 2012., 2nd Edition

COURSE LEARNING OUTCOMES (CO):

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

CLO1: Explain the fundamentals of computation, including regular languages, finite automata, and regular expressions.
CLO2: Describe context-free languages, pushdown automata, parsing techniques, and grammar normal forms.
CLO3: Analyze context-sensitive languages, Turing machines, and their computational models.
CLO4: Understand undecidability, computational complexity, and key decision problems in computation theory.

LEVEL OF CO-PO MAPPING TABLE

UNIT-WISE CLOs	**PLO 1**	**PLO 2**	**PLO 3**	**PLO 4**	**PL O5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**	**PLO1 0**	**Avg (CLO)**
MMCADTC225.1	3	2	0	0	0	0	0	0	0	0	**2.5**
MMCADTC225.2	0	0	3	2	0	0	0	0	0	0	**2.5**
MMCADTC225.3	0	0	0	0	3	2	0	0	0	0	**2.5**
MMCADTC225.4	0	0	0	0	0	0	3	3	0	0	**3.0**
Avg (PLO)	**1.5**	**1.0**	**1.5**	**1.0**	**1.5**	**1.0**	**1.5**	**1.5**	**0**	**0**	**2.88**

COURSE TITLE: Research and publication Ethics

Course Code: MMCACRP225						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practicals(P) :**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Learning Objectives

- To understand and apply fundamental principles of research ethics, including integrity, honesty, and responsibility, and analyze ethical dilemmas using established ethical frameworks and guidelines.
- To demonstrate knowledge of ethical publication practices, including authorship criteria, plagiarism prevention, and ethical responsibilities in peer review and citation.
- To utilize research tools to ensure ethical compliance, maintain data integrity, manage conflicts of interest, and evaluate ethical oversight processes through case analysis.
- To assess the societal impact of research, ethical considerations in emerging technologies, and global standards, while promoting transparency and responsible conduct in the evolving research landscape.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Research Ethics	**15 Hrs**
Introduction to research ethics, Research integrity and academic integrity, Scientific misconduct: falsification, fabrication, plagiarism, Moral philosophy: virtue ethics, deontology, consequentialism.	
UNIT 2: Scientific Conduct	**15 Hrs**
Ethical practices in research, Authorship and contributorship, Conflicts of interest, Responsibilities of a researcher, Research misconduct and handling allegations.	
UNIT 3: Publication Ethics Publication process and ethics, Redundant publication, salami slicing, Plagiarism and detection tools (Turnitin, iThenticate), Predatory journals and how to identify them, Ethics in peer review and editorial responsibility	**15 Hrs**
UNIT 4: Open Access Publishing and Copyright	**15 Hrs**
Types of open access: gold, green, hybrid, Creative Commons licenses, Copyright laws in research, Institutional repositories, Preprints and postprints, Impact factor, h-index, i10-index, Citation databases: Scopus, Web of Science, Google Scholar, Using reference management tools (Zotero, Mendeley, EndNote)	

Text Books

1. Research Ethics: A Practical Guide, Gary Comstock, Routledge, 2020.
2. Scientific Integrity: Text and Cases in Responsible Conduct of Research, Francis L. Macrina, ASM Press, 2014.

Reference Books

1. Publication Ethics: Rights and Wrongs in Academic Publishing, Norman K. Denzin and Michael D. Giardina, SAGE Publications, 2018.
2. The Ethics of Scientific Research: A Guidebook for Course Development, Judy E. Stern and Deni Elliott, University Press of New England, 1997.
3. Responsible Conduct of Research, Adil E. Shamoo and David B. Resnik, Oxford University Press, 2015.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Explain the fundamental principles, theories, and guidelines of research ethics and responsible conduct of research (RCR).
CLO2: Identify and analyze ethical issues in academic publishing, including authorship, peer review, plagiarism, and publication misconduct.
CLO3: Utilize ethical research tools and follow best practices for ensuring data integrity, transparency, and compliance with IRB protocols.
CLO4: Evaluate the societal and global impact of ethical research and interpret emerging ethical challenges in science and technology.

LEVEL OF CO-PO MAPPING TABLE

Unit wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCACRP225.1	2	2	1	2	1	3	2	3	1	2	**1.9**
MMCACRP225.2	2	2	2	2	1	3	2	3	2	3	**2.2**
MMCACRP225.3	2	2	3	3	3	2	2	2	2	3	**2.4**
MMCACRP225.4	1	2	2	2	2	3	3	3	2	3	**2.3**
Average (PLO)	**1.75**	**2.0**	**2.0**	**2.25**	**1.75**	**2.75**	**2.25**	**2.75**	**1.75**	**2.75**	**2.2**

COURSE TITLE: Mobile Application Development Lab

Course Code: MMCALMA225						**Examination Scheme**	**T**	**P**
Total number of Practical Hours: 60						**External**	**0**	**36**
						Internal	**0**	**14**
Lecture (L):	**0**	**Practical (P):**	**2**	**Tutorial (T):**	**0**	**Total Credits**		**2**

Course Learning Objectives:

- To develop proficiency in designing and implementing Android applications.
- To enable students to use Android Studio and SDK tools for application development.
- To understand and apply Android components like Activities, Intents, Broadcast Receivers, and Services.
- To design effective user interfaces using Android layouts and widgets.
- To integrate local databases and apply storage mechanisms for persistent data management.
- To apply object-oriented principles to mobile app development for modular and maintainable code.

Practical's

Week 1

- Set up Java development environment.
- Write and execute basic Java programs.
- Understand program structure and syntax.
- Learn to use the main method and print output to the console.

Week 2

- Learn to store data using arrays.
- Process data with loops (e.g., temperature tracking).
- Calculate averages and identify data points above a threshold.

Week 3

- Define user classes and create constructors.
- Initialize objects with values.
- Practice object creation and method invocation.

Week 4

- Use inheritance to model relationships (e.g., shapes).
- Override methods for specific behaviors in subclasses.
- Implement code reuse and flexible behavior.

Week 5

- Model a banking system with polymorphism and interfaces.
- Handle multiple account types through a common interface.
- Demonstrate code flexibility and extensibility.

Week 6

- Implement encapsulation in a student database.
- Use access modifiers (private, public) for data protection.
- Ensure data integrity and security through controlled access.

Week 7

- Create abstract classes and implement method overriding.
- Design game characters (e.g., Warrior, Wizard) with specific actions.
- Understand abstract classes for structuring game behaviors.

Week 8

- Install Android Studio or Eclipse with SDK.

- Set up the development environment for Android.
- Build and run a simple “Hello World” Android app.

Week 9

- Learn the key components of an Android app (Activity, Manifest, Layout).
- Modify project structure to understand component interaction.
- Understand how Android components work together in an app.

Week 10

- Navigate between Activities using Intents.
- Use Intent Filters to handle implicit Intents.
- Learn to send and receive data between components.

Week 11

- Declare app permissions in the Android Manifest.
- Request runtime permissions for sensitive features (e.g., camera, network).
- Understand Android’s security model and user privacy.

Week 12

- Create Broadcast Receivers to listen for system or app events.
- Send broadcasts to notify other components of events.
- Implement communication between different app components.

Week 13

- Design responsive UIs using LinearLayout, RelativeLayout, ConstraintLayout.
- Use UI components like TextView, Buttons, and ListViews.
- Ensure UIs are visually appealing and adaptable to screen sizes.

Week 14

- Implement advanced UI elements (animations, ScrollViews).
- Integrate SQLite for CRUD operations in an Android app.
- Create dynamic apps that store and retrieve data from local databases.

Textbooks

1. **Android Programming: The Big Nerd Ranch Guide** (5th Ed, 2022) – Bill Phillips et al.
2. **Head First Android Development** (3rd Ed, 2021) – Dawn & David Griffiths
3. **Professional Android** (4th Ed, 2018) – Reto Meier, Ian Lake

Reference Books

1. Kotlin Programming: The Big Nerd Ranch Guide (2019) – Josh Skeen
2. Mobile App Development with Flutter (2020) – Eric Windmill
3. Android Internals: A Confectioner's Cookbook – Jonathan Levin
4. Official Android Developer Guide – developer.android.com

COURSE LEARNING OUTCOMES (CO):

CLO1: Demonstrate basic Java programming and object-oriented concepts.

CLO2: Apply advanced OOP principles for secure and reusable code design.

CLO3: Set up Android environment and develop basic Android applications.

CLO4: Design responsive Android UIs and implement app communication and data storage.

LEVEL OF CO-PO MAPPING TABLE											
UNIT-WISE CLOs	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	PLO 10	**Avg (CLO)**
MMCALMA225.1	2	3	0	0	0	0	0	0	0	0	**2.5**
MMCALMA225.2	0	0	3	2	0	0	0	0	0	0	**2.5**
MMCALMA225.3	0	0	0	0	3	3	0	0	0	0	**3.0**
MMCALMA225.4	0	0	0	0	0	0	3	3	0	0	**3.0**
Avg (PLO)	**1.0**	**1.5**	**1.5**	**1.0**	**1.5**	**1.5**	**1.5**	**1.5**	**0**	**0**	**2.75**

SEMESTER-III

COURSE TITLE: Data Science with Python

Course Code: MMCACDS325						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**	

Course Objectives

- To introduce the foundational concepts of data analytics and data science, and develop an understanding of different types of analytics, their applications, and the skills required in a data-driven business environment.
- To equip students with practical knowledge of Python programming, including data structures, control flow, functions, and essential libraries for data analysis and visualization.
- To develop proficiency in preprocessing, cleaning, and transforming data using Python tools, with an emphasis on data wrangling, feature engineering, and handling various data formats.
- To enable students to perform exploratory data analysis, compute essential statistical measures, visualize data effectively, and interpret evaluation metrics relevant to classification and regression problems.

Course Content	**TEACHING HOURS**
UNIT I: Foundation of Data Analytics:	**15 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 II: Fundamentals of python	**15 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 III: Data Preprocessing and Wrangling	**15 Hrs.**
Introduction to data preprocessing, Data transformation-normalization, standardization, scaling, discretization, and binning. Encoding techniques – label encoding, one-hot encoding, ordinal encoding. Using Pandas for data manipulation – indexing, filtering, sorting, grouping, merging, reshaping, and pivoting datasets. Feature engineering – deriving new features, handling categorical and textual data	
UNIT IV: Exploratory Data Analysis and Model Evaluation Metrics	**15 Hrs.**

Concept and importance of Exploratory Data Analysis (EDA).Descriptive statistics – mean, median, mode, variance, standard deviation, percentiles, skewness, and kurtosis. Correlation analysis – Pearson, Spearman, and heatmaps.
Evaluation metrics for classification – Confusion matrix, Accuracy, Precision, Recall, F1 Score.Evaluation metrics for regression – Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

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

COURSE LEARNING OUTCOMES (CLO):

CLO1: Apply foundational concepts of data analytics and differentiate between descriptive, predictive, and prescriptive analytics with real-world applications.

CLO2: Demonstrate proficiency in Python programming by writing and executing code using built-in data structures, functions, and essential libraries such as Pandas, NumPy, and Matplotlib.

CLO3: Perform data preprocessing and wrangling tasks including handling missing values, encoding categorical variables, transforming and reshaping datasets for analysis.

CLO4: Conduct exploratory data analysis using statistical techniques and visualizations, and interpret evaluation metrics for assessing data quality and analytical outcomes.

CLO-PLO Matrix for the Course

	PLO										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCACDS325.1	3	1	2	1	2	1	2	1	2	2	**1.7**
MMCACDS325.2	2	3	3	1	2	0	2	1	1	3	**1.8**
MMCACDS325.3	2	3	3	1	3	0	2	2	2	3	**2.1**
MMCACDS325.4	2	3	3	2	3	0	1	2	2	3	**2.1**
Average (PLO)	**2.25**	**2.5**	**2.75**	**1.25**	**2.5**	**0.25**	**1.75**	**1.5**	**1.75**	**2.75**	**1.93**

<table>
<tr><th colspan="9">COURSE TITLE: Web Programming</th></tr>
<tr><td colspan="6"><b>Course Code: MMCACWP325</b></td><td><b>Examination Scheme</b></td><td><b>T</b></td><td><b>P</b></td></tr>
<tr><td colspan="6" rowspan="2"><b>Total number of Lecture Hours: 60</b></td><td><b>External</b></td><td><b>72</b></td><td><b>-</b></td></tr>
<tr><td><b>Internal</b></td><td><b>28</b></td><td><b>-</b></td></tr>
<tr><td><b>Lecture (L):</b></td><td>4</td><td><b>Practical (P):</b></td><td>-</td><td><b>Tutorial (T):</b></td><td>-</td><td colspan="2"><b>Total Credits</b></td><td><b>4</b></td></tr>
</table>

Course Objectives:

- To gain a comprehensive understanding of fundamental web technologies, including HTML, and CSS.
- To learn the principles of responsive and accessible web design using CSS and various layout techniques.
- To develop proficiency in JavaScript programming for client-side web development, including DOM manipulation and event handling.
- To acquire skills in server-side scripting using PHP to create dynamic and interactive web applications.
- To understand how to integrate and manage databases within web applications using MySQL.
- To combine client-side and server-side technologies to build complete, functional web applications.

Course Content	**TEACHING HOURS**
UNIT I: HTML and XHTML	**15 Hrs**
Introduction to HTML and XHTML: History and evolution of HTML and XHTML, Differences between HTML, Basic structure of an HTML document. HTML Basics: Elements and Attributes, creating paragraphs, headings, and lists, Working with images, links, and tables, Forms and form controls. Advanced HTML: Semantic HTML5 elements, Multimedia elements: audio and video	
UNIT II: CSS and Web Design	**15 Hrs**
Introduction to CSS: CSS syntax and selectors, Inline, internal, and external CSS, The cascade and inheritance. Styling Text and Elements: Fonts, text properties, and color, Styling lists, links, and tables, the box model: padding, margin, border Layout Techniques: Positioning elements: static, relative, absolute, and fixed, Flexbox and Grid layout systems, Responsive web design principles, Media queries for different devices.	
UNIT III: JavaScript	**15 Hrs**
Introduction to JavaScript: History and evolution of JavaScript, JavaScript syntax and data types, Variables, operators, and expressions. JavaScript Basics: Functions and scope, Control structures: loops and conditionals, Objects and arrays, The Document Object Model (DOM). JavaScript and the Web: Event handling, Form validation, Working with JSON, AJAX.	
UNIT IV: PHP and Server-Side Programming	**15 Hrs**
Introduction to PHP: History and features of PHP, Installing and configuring PHP, PHP syntax and data types. PHP Basics: Variables, constants, and operators, Control structures: conditionals and loops, Functions and arrays, Working with forms and user input. PHP and Databases: Connecting to a MySQL database, Performing CRUD operations, Prepared statements and security. Advanced PHP: Sessions and cookies.	
Textbooks	

1. Learning PHP, MySQL & JavaScript by Robin Nixon, O'Reilly Media 4th Edition (2018).
2. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett 1st Edition (2014)

Reference Books

1. Web Design The complete Reference, Thomas Powell, Tata McGrawHill 2nd Edition (2010)
2. HTML and XHTML The complete Reference, Thomas Powell, Tata McGrawHill 5th Edition (2010)
3. JavaScript 2.0: The Complete Reference, Second Edition by Thomas Powell and Fritz Schneider 2nd Edition (2004)
4. PHP: The Complete Reference by Steven Holzner, Tata McGrawHill 1st Edition (2008)

COURSE LEARNING OUTCOMES (CO):

CLO1: Students will be able to develop structured and interactive web pages using HTML and XHTML.
CLO2: Students will be able to apply CSS techniques to design responsive and visually appealing web layouts.
CLO3: Students will implement dynamic client-side functionality using JavaScript.
CLO4: Students will be able to build secure and dynamic server-side web applications using PHP and MySQL.

CLO-PLO Matrix for the Course

	PLO										
Unit-Wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCACWP325.1	2	3	2	1	2	1	1	2	1	2	**1.7**
MMCACWP325.2	2	3	3	1	2	1	1	3	1	2	**1.9**
MMCACWP325.3	2	3	3	1	3	1	1	3	1	3	**2.1**
MMCACWP325.4	2	3	3	1	3	2	1	2	1	3	**2.1**
Average (PLO)	**2.00**	**3.00**	**2.8**	**1.0**	**2.5**	**1.3**	**1.0**	**2.5**	**1.0**	**2.5**	**1.96**

COURSE TITLE: Quantum Computing

Course Code: MMCADQC325						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practicals (P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Learning Objectives

- To provide foundational knowledge of quantum computing principles, including quantum mechanics, qubits, and quantum gates, and their role in computational systems.
- To enable students to understand and analyze quantum algorithms, such as Shor's and Grover's algorithms, and their advantages over classical algorithms.
- To develop practical skills in quantum programming using tools like Qiskit and Cirq, enabling the design and simulation of quantum circuits.
- To identify and assess challenges in quantum computing, including hardware limitations, error correction, and scalability issues.
- To explore the application of quantum computing in fields like cryptography, optimization, and machine learning, and propose solutions for real-world problems

Course Content	TEACHING HOURS
UNIT 1: Introduction to Quantum Computing	**15 Hrs**
Fundamentals of Quantum Computing: Quantum mechanics principles (superposition, entanglement, measurement). Qubits: Representation, properties, and differences from classical bits. Quantum Gates: Single-qubit gates (Hadamard, Pauli), multi-qubit gates (CNOT, Toffoli).Quantum Circuits: Design and representation.Quantum Computing Models: Gate-based, adiabatic, and measurement-based quantum computing.Overview of Quantum Hardware: Superconducting qubits, trapped ions, and topological qubits.	
UNIT 2: Quantum Algorithms	**15 Hrs**
Introduction to Quantum Algorithms: Classical vs. quantum computational paradigms.Key Quantum Algorithms: Shor's algorithm (factoring), Grover's algorithm (search), Deutsch-Jozsa algorithm.Quantum Fourier Transform (QFT) and its applications.Algorithm Complexity: Speedup analysis and limitations.Algorithm Design Principles: Quantum parallelism, interference, and amplitude amplification.Simulation of Quantum Algorithms: Tools and techniques for algorithm testing.	
UNIT 3: Quantum Programming and Tools Quantum Programming Frameworks: Qiskit (IBM), Cirq (Google), PennyLane. Building Quantum Circuits: Syntax, libraries, and simulation environments. Quantum Programming Concepts: Quantum registers, measurements, and classical-quantum hybrid programs. Error Correction: Basics of quantum error correction codes (surface codes, stabilizer codes). Quantum Simulators vs. Real Quantum Hardware: Capabilities and limitations. Debugging and Optimization: Techniques for improving quantum circuit performance.	**14 Hrs**

UNIT 4: Applications and Future Trends	**14 Hrs**
Quantum Computing Applications: Cryptography (post-quantum cryptography), optimization problems, machine learning, and chemistry simulations.Quantum Computing in Industry: Use cases in finance, logistics, and healthcare. Challenges in Quantum Computing: Decoherence, noise, and scalability.Ethical and Security Implications: Impact on classical cryptography, data privacy, and responsible innovation. Emerging Trends: Quantum cloud computing (IBM Quantum, Amazon Braket), quantum supremacy, and quantum internet.Future Directions: Hybrid quantum-classical systems, fault-tolerant quantum computing, and quantum machine learning.	

Textbooks:

1. Quantum Computing: An Applied Approach, Jack D. Hidary, Springer, 2019.
2. Quantum Computation and Quantum Information, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press, 2010.

Reference Books

1. Learn Quantum Computing with Python and IBM Quantum Experience, Jack D. Hidary and Noson S. Yanofsky, Packt Publishing, 2020.
2. Quantum Computing for Computer Scientists, Eleanor G. Rieffel and Wolfgang H. Polak, Cambridge University Press, 2011.
3. Programming Quantum Computers, Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia, O'Reilly Media, 2019.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Understand the fundamental principles of quantum mechanics such as superposition, entanglement, and measurement, and differentiate classical computing from quantum computing models and hardware architectures.

CLO2: Analyze and interpret core quantum algorithms such as Shor's, Grover's, and Deutsch-Jozsa, and evaluate their performance in terms of computational complexity, speedup, and applicability.

CLO3: Develop and simulate quantum circuits and programs using modern quantum programming frameworks like Qiskit, Cirq, and PennyLane, and apply basic quantum error correction techniques.

CLO4: Evaluate the practical applications of quantum computing in areas such as cryptography, optimization, and machine learning, and discuss the ethical, security, and societal implications of quantum technologies and their future trends.

LEVEL OF CO-PO MAPPING TABLE

	PLOs										
Unit wise CLOs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADQC325.1	3	3	2	2	2	1	2	2	2	2	**2.1**

MMCADQC325.2	3	3	3	2	3	1	2	2	2	2	**2.3**
MMCADQC325.3	3	3	3	2	3	2	3	2	3	2	**2.6**
MMCADQC325.4	3	3	2	2	2	3	3	3	2	3	**2.6**
Average (PLO)	**3.0**	**3.0**	**2.5**	**2.0**	**2.5**	**1.75**	**2.5**	**2.25**	**2.25**	**2.25**	**2.4**

DCE-V

COURSE TITLE: Ethical Hacking

Course Code: MMCADEH325						Examination Scheme	T	P
Total number of Lecture Hours: 60						External	72	-
						Internal	28	-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits		4

Course Objectives

- To introduce students about ethical hacking, distinguishing it from malicious hacking, and covering hacker types, legal and ethical considerations, the hacking process, and essential tools, laying a foundation for advanced learning.
- To teach students network information gathering, scanning, and vulnerability assessment using tools like Nmap and Nessus, emphasizing passive/active reconnaissance and practical application for network security.
- To equip students with practical skills in exploiting vulnerabilities, securing networks, and protecting web applications using Kali Linux and tools like Metasploit.
- To equip students with techniques for covering tracks and maintaining anonymity, including log manipulation and the use of anonymity tools like VPNs and Tor

Course Content	TEACHING HOURS
UNIT 1: Introduction to Ethical Hacking	**15 Hrs.**
Definition and purpose, Differences between ethical hacking and malicious hacking, Importance of ethical hacking in cybersecurity, Types of Hackers (White Hat Hackers, Black Hat Hackers, Grey Hat Hackers), Legal and Ethical Considerations: Laws and regulations, Codes of conduct, Importance of permission and documentation, Ethical Hacking Process: Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks. Setting up your Kali Linux Environment Using a virtual environment, Navigating the Linux command line, Essential Linux Commands.	
UNIT 2: Reconnaissance and Scanning	**15 Hrs.**
Information Gathering: Passive vs. Active Reconnaissance, Tools for information gathering (e.g., Google Dorks, Whois), Network metadata analysis, tools for network metadata analysis, Active information gathering techniques: Nmap, Zenmap, Vulnerability Scanning: Understanding vulnerabilities (SQL Injection, XSS Attack), Tools: Nessus, OpenVAS, Enumeration: Identifying network resources and shares, Tools: Netcat, Nbtstat, Case Studies and Practical Examples: Performing a basic scan using Nmap and analyzing scan results.	
UNIT 3: Executing Vulnerability Assessment and Exploitation Techniques	**15Hrs.**
Exploiting Vulnerabilities: Exploitation frameworks in Kali Linux, Tools: Metasploit Framework, Password Attacks and Brute force: types of password attacks, Password cracking tools, Best practices and mitigations, Essential resources for understanding password attacks, Wireless networking fundamenta Types of wireless networks, Wireless network components, Basic Kali Linux commands for wireless Networking, Wireless network vulnerabilities and attacl Tools for wireless Network exploitation, Defending against wireless attacks.	
UNIT 4: Web Application Attacks, Covering Tracks and Reporting	**15 Hrs.**

Web application security fundamentals, Common web application components, Common web application threats, Understanding HTTP and HTTPS, Web Application Firewalls, scanning for vulnerabilities using Nitko, Brute forcing lo forms with Hydra, Exploiting SQL injection with sqlmap, Web application att and vulnerabilities, Web application exploitation tools and techniques, Covering Tracks: Importance of covering tracks, Techniques: Log manipulatic clearing logs, spoofing, Anonymity Tools: Proxy servers, VPNs, and Tor, Tools Proxy-Chains, Tor Browser, Reporting and Documentation: Importance of repoı in ethical hacking, Structure of a penetration testing report, writing an executive summary, and Creating a sample penetration testing report.

Textbooks

1. “Kali Linux for Ethical Hacking: Penetration testing and vulnerability assessment for network security Mohamed Atef, First Edition, BPB, 2024.
2. Network Security Assessment: Know Your Network by Chris McNab, 3rd Edition, Oreily, 2017.

Reference Books

1. Ethical Hacking and Penetration Testing Guide, by Rafay Baloch, CRC Press, 1e, 2015.

COURSE LEARNING OUTCOMES (CLO):

After completing the course, the student will be able to:

CLO1: Understand the purpose, legal context, and foundational phases of ethical hacking, and demonstrate basic proficiency with Kali Linux in a virtual environment.

CLO2: Apply passive and active reconnaissance, scanning, and enumeration techniques using industry-standard tools.

CLO3: Perform vulnerability assessments and exploit known vulnerabilities using tools like Metasploit, and understand wireless network attacks and defenses.

CLO4: Identify and exploit web application vulnerabilities, use tools like sqlmap and Hydra, and document findings through structured penetration testing reports.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADEH325.1	3	2	2	2	2	3	2	1	1	2	**2.0**
MMCADEH325.2	3	3	3	2	3	2	2	2	2	3	**2.5**
MMCADEH325.3	3	3	3	2	3	2	2	2	2	3	**2.5**
MMCADEH325.4	3	3	3	3	3	2	2	2	2	3	**2.6**
Average (PLO)	**3.0**	**2.75**	**2.75**	**2.25**	**2.75**	**2.25**	**2.0**	**1.75**	**1.75**	**2.75**	**2.4**

To be effective from year-2025

COURSE TITLE: Computer Vision

Course Code: MMCADCV325						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To Gain a comprehensive understanding of the principles, methodologies, and challenges in computer vision, including image formation, representation, and basic processing techniques.
- To develop the ability to extract meaningful features from images and apply advanced segmentation techniques, preparing for complex tasks like object recognition.
- To learn to apply computer vision algorithms for motion tracking, enabling the analysis and tracking of object movement in real-world video sequences.
- To acquire the skills to perform optical flow analysis, feature matching, and depth estimation, with a focus on 3D reconstruction using multi-camera systems.

Course Content	**TEACHING HOURS**
UNIT I: Introduction.	**15 Hrs**
Introduction to Computer Vision: Overview, History, and Applications of Computer Vision, Key Challenges, Image Formation and Representation, Basic Image Processing Techniques, Overview of Computer Vision Algorithms.	
UNIT II: Feature Extraction and Image Segmentation	**15 Hrs**
Feature Extraction: Edges (Canny, LOG, DOG), Line Detectors (Hough Transform), Corners (Harris and Hessian Affine), Orientation Histogram (SIFT, SURF, HOG, GLOH), Scale-Space Analysis (Image Pyramids and Gaussian Derivative Filters), Other Filters (Gabor Filters and DWT). Image Segmentation: Region Growing, Edge-Based Approaches to Segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation.	
UNIT III: Object Motion & Tracking	**15 Hrs**
Object Motion & Tracking: Understand Motion Models (Define Object Movement Over Time), Analyze Videos as Sequences of Individual Image Frames, Programmatically Track a Single Point Over Time, implement a Method for Tracking a Set of Unique Features Over Time.	
UNIT IV: Optical Flow & Feature Matching	**15 Hrs**
Optical Flow & Feature Matching: Optical Flow (Track a Moving Car Using Optical Flow), Feature Matching (Match Features from One Image Frame to Another), Depth Estimation and Multi-camera Views (Perspective, Binocular Stereopsis, Camera and Epipolar Geometry, Homography, Rectification, DLT, RANSAC, 3-D Reconstruction Framework, Auto-calibration).	

Textbooks:

1. Computer Vision: Algorithms and Applications" by Richard Zaleski, Springer Nature, 2nd Edition, 2022
2. Multiple View Geometry in Computer Vision" by Richard Hartley and Andrew Zisserman, Cambridge University Press, 2nd Edition, 2004
3. Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson Publishing, 4th Edition 2018.

Reference Books:

1. Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce, Pearson Publishing, 2nd Edition, 2012.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Students will describe the history, applications, and challenges in computer vision.

CLO2: Students will extract features using methods like SIFT, SURF, HOG, and Hough Transform.

CLO3: Students will model object motion and track features over time across video frames.

CLO4: Students will apply optical flow and feature-matching algorithms to image sequences.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	**PLO 1**	**PLO 2**	**PLO 3**	**PLO 4**	**PLO 5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**	**PLO 10**	**Average (CLO)**
MMCADCV325.1	3	1	1	1	2	0	1	1	1	2	**1.3**
MMCADCV325.2	2	3	2	1	3	0	1	2	2	3	**1.9**
MMCADCV325.3	2	3	3	1	3	0	1	2	2	3	**2.0**
MMCADCV325.4	2	3	2	1	3	0	1	2	2	3	**1.9**
Average (PLO)	**2.3**	**2.5**	**2.0**	**1.0**	**2.8**	**0.0**	**1.0**	**1.8**	**1.8**	**2.8**	**1.78**

COURSE TITLE: Enterprise Resource Planning

Course Code: MMCADER325						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To understand the evolution, definition, and growth of ERP, including its advantages, various modules, and relevance to different business models, particularly in the context of India.
- To analyse the relationship between ERP and related technologies, such as Business Process Reengineering (BPR), Management Information Systems (MIS), and Supply Chain Management (SCM).
- To evaluate the ERP implementation lifecycle, including planning, system selection, training, data migration, and the roles of consultants, vendors, and employees in successful ERP implementation.
- To assess the post-implementation challenges of ERP systems, the factors influencing their success or failure, and explore emerging trends such as extended ERP systems, CRM, SCM, and web-enabled ERP solutions.

Course Content	**TEACHING HOURS**
UNIT I: Introduction to ERP	**15 Hrs**
Evolution of ERP; what is ERP? Reasons for the Growth of ERP; Scenario and Justification of ERP in India; Evaluation of ERP; Various Modules of ERP; Advantage of ERP. An Overview of Enterprise. Integrated Management Information; Business Modeling; ERP for Small Business; ERP for Make to Order Companies; Business Process Mapping for ERP Module Design; Hardware Environment and its Selection for ERP Implementation	
UNIT II: ERP and related Technologies	**15 Hrs**
ERP and Related Technologies; Business Process Reengineering (BPR); Management Information System (MIS); Executive Information System (EIS); Decision support System (DSS); Supply Chain Management (SCM	
UNIT III: ERP Implementation	**15 Hrs**
ERP Implementation: Planning Evaluation and selection of ERP systems - Implementation life cycle - ERP implementation, Methodology and Frame work-Training – Data Migration - People Organization in implementation-Consultants, Vendors and Employees	
UNIT IV: Post Implementation and Emerging Trends	**15 Hrs**
Post Implementation: Maintenance of ERP- Organizational and Industrial impact; Success and Failure factors of ERP Implementation Emerging Trends on ERP: Extended ERP systems and ERP add-ons -CRM, SCM, Business analytics - Future trends in ERP systems-web enabled.	

Textbooks

1. Manufacturing Resource Planning (MRP II) with Introduction to ERP; SCM; an CRM by Khalid Sheikh, Publisher: McGraw-Hill
2. ERP and Supply Chain Management by Christian N. Madu, Publisher: CHI.

Reference Books

1. Jagan Nathan Vaman, ERP in Practice, Tata McGraw-Hill, 2008
2. Alexis Leon, Enterprise Resource Planning, second edition, Tata McGraw-Hill, 2008.
3. Mahadeo Jaiswal and Ganesh Vanapalli, ERP Macmillan India, 2009
4. Vinod Kumar Grag and N.K. Venkitakrishnan, ERP- Concepts and Practice, PHI.

COURSE LEARNING OUTCOMES (CLO):

By the end of this course, students will be able to:

CLO1: Explain the evolution, significance, and advantages of ERP systems, and evaluate ERP modules and their applicability to different business environments, including small and make-to-order enterprises.
CLO2: Analyze the relationship between ERP systems and associated technologies such as BPR, MIS, EIS, DSS, and SCM, and assess their role in enhancing organizational efficiency.
CLO3: Demonstrate understanding of ERP implementation life cycles, methodologies, and frameworks, including vendor and consultant roles, training, and data migration processes.
CLO4: Evaluate the post-implementation impact of ERP systems on organizations, identify critical success and failure factors, and explore emerging trends like CRM, SCM integration, and web-enabled ERP systems.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	PLO 10	Avg (CLO)
MMCADER325.1	3	2	3	2	2	1	1	2	1	2	**1.90**
MMCADER325.2	2	3	3	2	3	1	2	2	2	3	**2.30**
MMCADER325.3	3	3	3	2	3	2	2	2	2	3	**2.50**
MMCADER325.4	3	2	3	3	3	2	2	3	2	3	**2.60**
Average (PLO)	**2.75**	**2.50**	**3.00**	**2.25**	**2.75**	**1.50**	**1.75**	**2.25**	**1.75**	**2.75**	**2.32**

DCE-VI

COURSE TITLE: Natural Language Processing

Course Code: MMCADNL325		**Examination Scheme**	
Total number of Lecture Hours: 60		**External**	**72**
		Internal	**28**

Lecture (L):	**4**	**Practicals(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Learning Objectives

- To provide foundational knowledge of natural language processing principles, including text representation, linguistic structures, and computational models.
- To enable students to understand and analyze key NLP algorithms, such as tokenization, part-of-speech tagging, and sentiment analysis, and their role in text processing.
- To develop practical skills in NLP programming using tools like NLTK, spaCy, and Transformers, enabling the design and implementation of NLP pipelines.
- To identify and assess challenges in NLP, including data sparsity, bias in language models, and multilingual processing.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to Natural Language Processing	**15 Hrs**
Introduction to NLP: definitions, goals, applications, Components of NLP: tokens, lexicon, grammar, semantics. Text preprocessing techniques: tokenization, normalization, stop-word removal, stemming vs. lemmatization, Basic feature extraction: bag-of-words, TF–IDF.	
UNIT 2: Linguistic Analysis	**15 Hrs**
Part-of-Speech tagging and morphological analysis, Syntactic parsing: constituents vs. dependency trees, Chunking and shallow parsing, Syntax-based feature generation, Evaluating tagging and parsing performance	
UNIT 3: Semantics & Language Modeling Word semantics: embeddings (Word2Vec, GloVe), Contextual representations (e.g., BERT overview), Language modeling: n-grams and neural models, Word sense disambiguation applications, Semantic similarity and clustering	**15Hrs**
UNIT 4: Applications and Future Trends	**15 Hrs**
Sentiment analysis and opinion mining , Text classification and document clustering, Information extraction: NER, relation extraction, Introduction to question answering and summarization, Advanced interpretability (e.g. feature importance, opening the door to your SHAP work)	

Textbooks:

1. Speech and Language Processing, Daniel Jurafsky and James H. Martin, Prentice Hall, 2020.

2. Deep Learning for Natural Language Processing, Palash Goyal, Sumit Pandey, and Karan Jain, Apress, *2018.*

Reference Books

1. Natural Language Processing with Python, Steven Bird, Ewan Klein, and Edward Loper, O'Reilly Media, 2009.
2. Foundations of Statistical Natural Language Processing, Christopher D. Manning and Hinrich Schütze, MIT Press, 1999.
3. Transformers for Natural Language Processing, Denis Rothman, Packt Publishing, 2021.

COURSE LEARNING OUTCOMES (CLO):

CLO1: Explain the foundational concepts of natural language processing, including text representation, preprocessing, and word embeddings.
CLO2: Analyze and apply key NLP algorithms such as HMMs, CRFs, RNNs, LSTMs, and transformer-based models.
CLO3: Develop NLP pipelines using standard programming libraries and frameworks (e.g., NLTK, spaCy, Hugging Face).
CLO4: Evaluate real-world NLP applications, assess challenges like bias and multilinguality, and explore emerging trends like LLMs and conversational AI.

LEVEL OF CO-PO MAPPING TABLE

Unit wise CLOs	PLOs									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg (CLO)**
MMCADNL325.1	3	2	2	2	2	1	1	2	1	2	**1.8**
MMCADNL325.2	3	3	3	2	3	1	2	2	2	2	**2.3**
MMCADNL325.3	3	3	3	2	3	2	2	2	2	2	**2.4**
MMCADNL325.4	3	3	2	2	2	3	3	3	2	3	**2.6**
Average (PLO)	**3.0**	**2.75**	**2.5**	**2.0**	**2.5**	**1.75**	**2.0**	**2.25**	**1.75**	**2.25**	**2.28**

COURSE TITLE: Software Quality Assurance

Course Code: MMCADSQ325						**Examination Scheme**	**T**	**P**	
Total number of Lecture Hours: 60						**External**	**72**	**-**	
						Internal	**28**	**-**	
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**	

Course Objectives

- To understand the components of the Software Quality Assurance System, including Pre-Project Software Quality Components, Contract Review, and Development and Quality Plans.
- To apply various quality engineering tools and techniques, such as the Seven Basic Quality Tools, Statistical Process Control, and Failure Mode and Effect Analysis (FMEA).
- To analyze different software quality standards and models, including ISO 9000 series, CMMI, Six Sigma, and the integration of AI techniques in software testing and quality assurance.
- To implement Scrum in an organization, understanding the steps for transitioning to Scrum, Scrum artifacts, and the use of Agile project management tools like JIRA and Trello.

Course Content	**TEACHING HOURS**
UNIT I: Software Quality	**15 Hrs**
Definition of Software Quality, Quality Planning, Quality system. Quality Control vs Quality Assurance , Product life cycle , Project life cycle models. The Software Quality Challenge, Software Quality Factors, Components of the Software Quality Assurance System. Pre-Project Software Quality Components, Contract Review, Development and Quality Plans	
UNIT II: Software Quality Engineering Tools And Techniques	**15 Hrs**
Supporting quality activities- Metrics, Reviews, SCM , Software quality assurance and risk management, Seven basic Quality tools, Checklist, Pareto diagram, Cause and effect diagram, Run chart, Histogram, Control chart, Scatter diagram, Poka Yoke, Statistical process control ,Failure Mode and Effect Analysis, Quality Function deployment, Continuous improvement	
UNIT III: Quality Assurance Models and AI Assurance	**15 Hrs**
Software Quality Standards, ISO 9000 series, CMM, CMMI, P-CMM , Six Sigma, Malcolm Baldrige Quality, Introduction to AI in Software Quality Assurance-Definition and Importance of AI Assurance, Overview of AI in Software Development and Testing, Benefits and Challenges of Integrating AI in QA, Overview of AI Techniques in Software Testing, Automated Test Case Generation-Techniques for Automated Test Case Generation, Ensuring AI Model Quality-Verification and Validation of AI Models, Testing AI Models for Accuracy, Robustness, and Fairness	
Unit IV: Scrum and Agile Model	**15 Hrs**
Introduction to Agile-Definition and History of Agile, Principles of Agile Manifesto, Benefits of Agile Methodology, Overview of Scrum Framework-Definition of Scrum, Scrum Values and Principles, Scrum Artifacts-Product Backlog, Sprint Backlog, Increment, Agile Project Management Tools-Introduction to Tools (e.g., JIRA, Trello, Asana), Implementing Scrum in an Organization-Steps to Transition to Scrum	
Textbooks:	

1. Galin, D. (2018). Software Quality Assurance: From Theory to Implementation (2nd ed.). Pearson.

2. Godbole, N. S. (2017). Software Quality Assurance: Principles and Practice (1st ed.). Alpha Science International Ltd.

Reference Books:

1. Cohn, M. (2006). Agile Estimating and Planning. Prentice Hall.
2. Partridge, D. (1992). Artificial Intelligence in Software Engineering. Routledge.
3. Black, R. (2002). Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing (2nd ed.). Wiley.
4. Jones, C. (2017). Applied Software Measurement: Global Analysis of Productivity and Quality (3rd ed.). Addison-Wesley Professional.

COURSE LEARNING OUTCOMES (CO):

CLO1: Understand software quality principles, quality planning, assurance components, and their relevance across software and project life cycles.

CLO2: Apply quality engineering tools and risk management strategies to ensure continuous software quality improvement.

CLO3: Analyze software quality standards and evaluate AI-assisted quality assurance methods and model validation techniques.

CLO4: Demonstrate the ability to implement Agile principles and Scrum framework using modern project management tools.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	**PLOs**										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average (CLO)**
MMCADSQ325.1	3	2	3	1	3	3	2	2	2	3	**2.4**
MMCADSQ325.2	3	3	3	1	3	3	2	2	2	3	**2.5**
MMCADSQ325.3	3	2	3	1	3	2	2	2	3	3	**2.4**
MMCADSQ325.4	2	3	3	2	3	2	2	2	2	3	**2.4**
Average (PLO)	**2.75**	**2.5**	**3.0**	**1.25**	**3.0**	**2.5**	**2.0**	**2.0**	**2.25**	**3.0**	**2.43**

COURSE TITLE: Deep Learning

Course Code: MMCADDL325						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**		**4**

Course Objectives

- To provide a comprehensive understanding of deep learning principles, including the distinction between shallow and deep architectures.
- To equip students with the skills to design, implement, and train artificial neural networks (ANNs) and convolutional neural networks (CNNs) for various applications.
- To introduce students to advanced deep learning architectures, techniques, and challenges, including regularization, transfer learning, and neural architecture search.
- To explore cutting-edge topics in deep learning, such as Graph Neural Networks, Meta Learning, Auto encoders, Generative Adversarial Networks (GANs), and Deep Reinforcement Learning.

Course Content	**TEACHING HOURS**
UNIT I: Fundamentals of Neural Networks and Deep Learning	**15 Hrs**
Introduction to Machine Learning vs Deep Learning, Biological Neuron vs Artificial Neuron, Basics of Neural Networks: Perceptron, Multilayer Perceptron (MLP), Activation Functions: Sigmoid, Tanh, ReLU, Leaky ReLU, Softmax, Loss Functions: MSE, Cross-Entropy, Gradient Descent and Backpropagation, Overfitting and Underfitting, Bias-Variance Tradeoff	
UNIT II: Deep Neural Networks and Training Techniques	**15 Hrs**
Deep Neural Network Architectures, Batch Normalization and Dropout, Weight Initialization Methods, Optimization Algorithms: SGD, Adam, RMSProp, Hyperparameter Tuning: Epochs, Batch Size, and Learning Rate, Model Evaluation: Confusion Matrix, Precision, Recall, F1 Score, ROC-AUC	
UNIT III : Convolution Neural Networks	**15 Hrs**
Introduction to CNNs: Need and Applications, CNN Architecture: Convolutional Layers, Pooling Layers, Fully Connected Layers, Filters, Stride, Padding, Transfer Learning and Pre-trained Models (VGG, ResNet, etc.), Data Augmentation, Case Studies: Image Classification, Object Detection (basic intro)	
UNIT IV: Advanced Topics and Applications	**15 Hrs**
Recurrent Neural Networks (RNNs): Basics and Applications, Long Short-Term Memory (LSTM) and GRU, Introduction to Generative Adversarial Networks (GANs), Autoencoders and Variational Autoencoders, Applications of Deep Learning in NLP, Computer Vision, and Healthcare	
Textbooks	

1. Deep Learning by Ian GoodFellow, MIT Press.2016
2. Advanced Deep Learning with Python, Ivan Vasilev, 2019
3. Advances in Deep Learning, M. Arif Wani, 2019

Reference Books

1. Deep Learning with Python, Francois Chollet, 2nd edition, 2021
2. Deep Reinforcement Learning Hands-On, Maxim Lapan, 2nd edition, 2020
3. Automated Machine Learning Methods, Systems, Challenges, 2019
4. Deep Learning: A Visual Approach, Andrew Glassner, 2021
5. Selected Journal and Conference Papers.

COURSE LEARNING OUTCOMES (CO):

CLO1: Understand core concepts of deep learning and artificial neural networks.

CLO2: Apply convolutional neural network techniques and analyze their architectures.

CLO3: Evaluate neural architecture search methods and their challenges.

CLO4: Implement and assess advanced deep learning models in real-world applications.

LEVEL OF CO-PO MAPPING TABLE

UNIT-WISE CLOs	**PLO 1**	**PLO 2**	**PLO 3**	**PLO 4**	**PLO 5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**	**PLO 10**	**Avg (CLO)**
MMCADDL325.1	3	2	2	0	0	0	0	0	0	0	**2.33**
MMCADDL325.2	0	0	3	3	2	0	0	0	0	0	**2.67**
MMCADDL325.3	0	0	0	0	3	2	0	0	0	0	**2.5**
MMCADDL325.4	0	0	0	0	0	0	3	3	0	0	**3.0**
Avg (PLO)	**1.5**	**1.0**	**1.67**	**1.5**	**1.5**	**1.0**	**1.5**	**1.5**	**0**	**0**	**2.63**

COURSE TITLE: Internet of Things

Course Code: MMCADIT325						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**		**4**

Course Objectives

- To understand the fundamental concepts of cloud computing, including different cloud deployment models (public, private, hybrid) and service models (IaaS, PaaS, SaaS).
- To explore the technologies and processes involved in deploying and managing cloud-based applications and web services.
- To analyze the management aspects of cloud services, including reliability, availability, scalability, and the economic factors influencing cloud platform choices.
- To gain practical knowledge in cloud-based application development and service creation environments, focusing on the benefits and challenges of cloud architecture.

Course Content	**TEACHING HOURS**
UNIT 1:	**15 Hrs**
Definition & Characteristics of Iot, Physical Design of Iot, Things in Iot, Iot Protocols; Logical Design Of Iot: Iot Functional Blocks, Iot Communication. Models, Iot Communication APIs; IoT Levels and Templates. Domain. Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle.	
UNIT 2:	**15 Hrs**
Wireless Sensor Networks, Cloud Computing, Big Data Analytic, Communication Protocols, Machine to Machine, Difference between IoT and M2M, Software define Network, Embedded Systems. Design challenges, Development challenges, Security challenges, Other challenges.	
UNIT 3:	**15 hrs**
Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraints-hardware is popular again. Data representation and visualization, Interaction and remote control. Industrial Automation- Service oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-AESOP: from the Web of Things to the Cloud of Things.	
UNIT 4 :	**15 hrs**
Setting up the Arduino development environment: Options for Internet connectivity, Interacting with basic sensors, Interacting with basic actuators, Configuring Arduino for the IoT. Grabbing the content from a web page, Sending data to the cloud, Monitoring sensor data from a cloud dashboard, Monitoring several Arduino boards at, Storing data on Google Drive. Basic local M2M interactions, Cloud M2M with IFTTT; Case Study: IoT based Flood Monitoring and Alert System.	
Textbooks	

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014. (ISBN-13: 978-8173719547)
2. Schwartz, Marco. “Internet of Things with Arduino Cookbook”. Packt Publishing Ltd, 2016.

Reference Books

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of
2. Intelligence”, 1st Edition, Academic Press, 2014. (ISBN-13: 978-0124076846)
3. The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World, Michael Miller

COURSE LEARNING OUTCOMES (CLO):

CLO1: Students will be able to describe and differentiate between key IoT enabling technologies, such as Wireless Sensor Networks, Cloud Computing, and M2M communication, while understanding the role of software-defined networks and embedded systems.

CLO2: Students will gain knowledge of the challenges in IoT design and development, including security and technical constraints, and propose solutions to overcome these issues.

CLO3: Students will acquire skills in designing and analyzing IoT architecture, considering various views like functional, information, and deployment, and apply these concepts to industrial automation and real-world applications.

CLO4: Students will develop practical skills in using Arduino for IoT projects, including setting up the environment, connecting to sensors and actuators, and implementing cloud-based data monitoring and M2M interactions.

LEVEL OF CLO-PLO MAPPING TABLE

CLOs	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10	Average (CLO)
MMCADIT325.1	3	1	2	0	2	1	1	1	1	2	1.4
MMCADIT325.2	2	1	2	1	3	3	2	2	2	3	2.1
MMCADIT325.3	2	2	3	1	3	1	2	2	2	3	2.1
MMCADIT325.4	2	3	3	1	3	1	2	3	2	3	2.3
Average(PLO)	2.25	1.75	2.5	0.75	2.75	1.5	1.75	2	1.75	2.75	1.97

COURSE TITLE: Software Project Management

Course Code: MMCACSP325						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practicals(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**

Course Learning Objectives

- To provide fundamental skills of software Project management emphasizing on issues & hurdles associated with delivering successful projects.
- To apply project management concepts through working in a group as team leader or active team member on an IT project.
- To utilize scheduling terminology, techniques, and tools to create accurate and feasible project timelines.
- To identify potential risks in software projects and develop mitigation strategies to minimize their impact on project outcomes.
- To define and measure key performance indicators (KPIs) to assess project success, including quality, scope adherence, and stakeholder satisfaction.

Course Content	**TEACHING HOURS**
UNIT 1: Introduction to SPM	**15 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	**15 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	**15 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	**15 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 LEARNING OUTCOMES (CLO):

CLO1: Understand the foundational concepts of Software Project Management, including project lifecycle, planning, and estimation techniques.

CLO2: Apply scheduling techniques (WBS, PERT, CPM, Gantt Charts) and organize personnel effectively in a software project environment.

CLO3: Analyze and apply project monitoring methods like Earned Value Analysis (EVA) and software reviews to ensure project control.

CLO4: Evaluate software quality using CMM, quality attributes, and formal SQA approaches; understand contracts and quality metrics in software projects.

LEVEL OF CLO-PLO MAPPING TABLE

Unit wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCACSP325.1	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCACSP325.2	3	3	3	2	3	1	2	3	2	3	**2.5**
MMCACSP325.3	3	3	3	2	3	1	2	2	2	3	**2.4**
MMCACSP325.4	3	3	3	2	3	2	2	2	2	3	**2.5**
Average (PLO)	**3.0**	**3.0**	**3.0**	**2.0**	**3.0**	**1.25**	**2.0**	**2.25**	**2.0**	**3.0**	**2.5**

COURSE TITLE: Data Science with Python Lab

						Examination Scheme	T	P
Course Code: MMCALDS325								
Total number of Practical Hours: 60						**External**	**0**	**36**
						Internal	**0**	**14**
Lecture (L):	**0**	**Practical (P):**	**2**	**Tutorial (T):**	**0**	**Total Credits**		**2**

Course Learning Objectives:

- To gain hands-on experience with Python programming syntax, data types, and control structures.
- To apply functional, modular, and object-oriented programming concepts to solve computational problems.
- To utilize Python libraries such as NumPy, Pandas, Matplotlib, and SciPy for data manipulation and analysis.
- To implement and understand the basics of machine learning algorithms using Python on real-world datasets.

Practical's

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 LEARNING OUTCOMES (CO):

CLO1: Students will be able to write and execute basic Python programs using fundamental concepts such as variables, data types, operators, conditional statements, and loops.

CLO2: Students will apply functions, modules, packages, and object-oriented programming principles (classes, inheritance, and polymorphism) to develop modular and reusable Python programs.

CLO3: Students will use Python libraries such as NumPy, Pandas, Matplotlib, and SciPy to perform data manipulation, visualization, and statistical analysis.

CLO4: Students will implement basic machine learning and decision-making techniques such as regression analysis and decision tree classification using Python.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	**PLO 1**	**PLO 2**	**PLO 3**	**PLO 4**	**PLO 5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**	**PLO 10**	**Avg (CLO)**
MMCALDS325.1	2	3	2	1	2	0	1	1	0	2	**1.4**
MMCALDS325.2	2	3	2	1	2	0	1	2	1	2	**1.6**
MMCALDS325.3	2	3	3	1	3	0	2	2	2	3	**2.1**
MMCALDS325.4	2	2	3	1	3	0	1	2	2	3	**1.9**
Average (PLO)	**2.0**	**2.75**	**2.5**	**1.0**	**2.5**	**0.0**	**1.25**	**1.75**	**1.25**	**2.5**	**1.75**

COURSE TITLE: Web Programming Lab

Course Code: MMCALWP325						**Examination Scheme**	**T**	**P**
Total number of Practical Hours: 60						**External**	**-**	**36**
						Internal	**-**	**14**
Lecture (L):	**-**	**Practical (P):**	**2**	**Tutorial (T):**	**-**	**Total Credits**		**2**

Course Objectives:

- **To introduce students about the foundational concepts of web development** through hands-on experience with **HTML, CSS, JavaScript, PHP, and MySQL**.
- **To enable students about the structure and design static web pages** using HTML and CSS, including text formatting, layout design, multimedia, forms, and semantic elements.
- **To teach students to apply styling techniques** through various types of CSS, explore the box model, and implement responsive design with media queries.
- **To provide a working understanding of JavaScript for client-side interactivity**, including functions, loops, conditionals, DOM manipulation, and event handling.
- **To familiarize students with basic programming in PHP**, including variables, control structures, arrays, functions, and interaction with HTML forms.

Practical's

Week 1

- Create a basic HTML document with paragraphs, headings, and lists.
- Add images, links, and tables to an HTML document.
- Create a simple form with various form controls (text inputs, radio buttons, checkboxes, etc.).

Week 2

- Create an HTML document using semantic HTML5 elements.
- Validate an XHTML document and correct any errors.
- Convert an HTML document to XHTML and ensure it follows proper syntax rules.

Week 3

- Create a basic CSS file and link it to an HTML document.
- Apply inline, internal, and external CSS styles to a webpage.
- Use CSS selectors to style different HTML elements.

Week 4

- Style lists, links, and tables using CSS.
- Implement the box model: padding, margin, and border.

Week 5

- Create a simple webpage layout using static, relative, absolute, and fixed positioning.

- Create a responsive webpage using media queries.

Week 6

- Implement CSS animations and transactions on a webpage.
- Write a simple JavaScript program that uses variables, operators, and expressions.

Week 7

- Create JavaScript functions and demonstrate scope.
- Implement control structures (loops and conditionals) in JavaScript.

Week 8

- Create and manipulate JavaScript objects and arrays.
- Use the Document Object Model (DOM) to interact with an HTML document.

Week 9

- Write a JavaScript program to handle events on a webpage.
- Write a JavaScript program that handles errors and debugging.

Week 10

- Write a basic PHP script to output “Hello, World!”.
- Create a PHP script that uses variables, constants, and operators.
- Write PHP programs using control structures (conditionals and loops).

Week 11

- Create PHP functions and work with arrays.
- Develop a simple form in PHP and handle user input.
- Connect a PHP script to a MySQL database.

Week 12

- Perform CRUD (Create, read, Update, delete) operations in PHP.
- Implement session management in PHP.

Week 13

- Use cookies in PHP to store user preferences.
- Handle errors and debug a PHP application.

Week 14

- Develop the frontend using HTML, CSS and JavaScript.
- Create the backend using PHP and integrate it with MySQL database.

COURSE LEARNING OUTCOMES (CO):

By the end of this practical course, students will be able to:

CLO1: Students will be able to create well-structured documents using paragraphs, headings, lists, tables, and forms.
CLO2: Students will demonstrate the ability to use CSS for styling text, links, tables, and layouts using box model and positioning.
CLO3: Students will write JavaScript programs using variables, functions, control structures, and arrays.
CLO4: Students will write PHP scripts with form handling, session management, and database connectivity.

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	PLO 10	Average (CLO)
MMCALWP325.1	2	3	2	1	2	1	1	2	1	2	**1.7**
MMCALWP325.2	2	3	3	1	2	1	1	3	1	2	**1.9**
MMCALWP325.3	2	3	3	1	3	1	1	3	1	3	**2.1**
MMCALWP325.4	2	3	3	1	3	2	1	2	1	3	**2.1**
Average (PLO)	**2.00**	**3.00**	**2.8**	**1.0**	**2.5**	**1.3**	**1.0**	**2.5**	**1.0**	**2.5**	**1.96**

SEMESTER-IV

PROJECT:

PROBLEM IDENTIFICATION & ANALYSIS

PROJECT:

DISSERTATION

PROJECT:

SOFTWARE DEVELOPMENT

PROJECT:

RESEARCH COMPONENT

Table: CLOs-PLOs Mapping Matrix for all the courses

COURSE CODE	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9	PLO 10	Average CLO
MMCACJP125	2.25	2.75	2.25	1.25	2.75	1	1.5	1.75	1.25	2.75	**1.95**
MMCACML125	3.0	2.75	2.75	2.0	2.75	1.0	2.0	2.0	2.0	2.75	**2.3**
MMCADAD125	3.0	3.0	2.5	1.0	3.0	0.25	2.0	2.5	1.5	3	**2.2**
MMCADCG125	1.5	1.67	1.5	1.0	1.5	1.5	1.5	1.5	0	0	**2.79**
MMCADMI125	2.25	2.25	2.25	2.25	2.25	2.25	1.5	2.0	1.75	2.5	**2.1**
MMCADSE125	3.0	3.0	3.0	2.0	2.75	1.0	2.0	2.25	2.0	3.0	**2.4**
MMCADDS125	3.0	2.5	2.3	1.0	2.8	1.3	1.0	1.8	1.5	2.5	**1.97**
MMCADAI125	2.75	2.00	2.25	1.25	2.25	1.00	1.25	2.00	1.25	2.25	**1.82**
MMCADBC125	3.0	2.5	2.25	2.0	3.0	1.75	2.25	2.5	2.5	2.75	**2.45**
MMCADCS125	3.0	2.5	2.8	1.0	3.0	2.75	2.0	2.25	1.75	3.0	**2.4**
MMCACRM125	3.0	2.5	2.5	2.5	3.0	2.5	3	2	3	2	**2.6**
MMCALJP125	2.5	3	2.75	1.25	2.75	0.5	1.25	2	2	1.5	**1.95**
MMCALML125	3.0	2.75	2.75	2.0	2.75	1.0	2.0	2.0	2.0	2.75	**2.3**
MMCACDA225	2.8	2.5	2.5	0.25	3.0	0.25	1.0	1.5	1.75	3.0	**1.85**
MMCACMA225	1.5	1.0	1.67	1.5	2.0	1.5	1.0	1.5	1.0	0	**2.63**
MMCADAO225	1.5	1.0	1.67	1.5	2.5	2.0	1.5	1.5	0	0	**2.54**
MMCADDI225	2.5	2.5	2.5	0.75	2.75	0.5	1.5	2	1	2.75	**1.8**
MMCADDS225	3.0	2.5	2.5	2.25	2.75	1.25	2.25	2	2	2.5	**2.3**
MMCADCN225	2.75	2.75	1.75	1.25	2.5	2.25	1.25	1.5	1.5	2.5	**2**
MMCADAC225	2.75	2.25	2.25	1.75	1.75	1.25	1.5	1.25	1.25	2.25	**1.38**
MMCADCC225	2.25	2.25	2.5	0.75	2.5	1	1.5	1.75	1	2.75	**1.8**
MMCADLP225	2.75	3.0	3.0	1.25	3.0	1.75	2.0	2.75	1.25	3.0	**2.38**
MMCADTC225	1.5	1.0	1.5	1.0	1.5	1.0	1.5	1.5	0	0	**2.88**
MMCACRP225	1.75	2.0	2.0	2.25	1.75	2.75	2.25	2.75	1.75	2.75	**2.2**
MMCALMA225	1.0	1.5	1.5	1.0	1.5	1.5	1.5	1.5	0	0	**2.75**
MMCACDS325	2.25	2.5	2.75	1.25	2.5	0.25	1.75	1.5	1.75	2.75	**1.93**
MMCACWP325	2.00	3.00	2.8	1.0	2.5	1.3	1.0	2.5	1.0	2.5	**1.96**
MMCADQC325	3.0	3.0	2.5	2.0	2.5	1.75	2.5	2.25	2.25	2.25	**2.4**
MMCADEH325	3.0	2.75	2.75	2.25	2.75	2.25	2.0	1.75	1.75	2.75	**2.4**
MMCADCV325	2.3	2.5	2.0	1.0	2.8	0.0	1.0	1.8	1.8	2.8	**1.78**
MMCADER325	2.75	2.50	3.00	2.25	2.75	1.50	1.75	2.25	1.75	2.75	**2.32**
MMCADNL325	3.0	2.75	2.5	2.0	2.5	1.75	2.0	2.25	1.75	2.25	**2.28**
MMCADSQ325	2.75	2.5	3.0	1.25	3.0	2.5	2.0	2.0	2.25	3.0	**2.43**
MMCADDL325	1.5	1.0	1.67	1.5	1.5	1.0	1.5	1.5	0	0	**2.63**
MMCADIT325	2.25	1.75	2.5	0.75	2.75	1.5	1.75	2	1.75	2.75	**1.97**
MMCACSP325	3.0	3.0	3.0	2.0	3.0	1.25	2.0	2.25	2.0	3.0	**2.5**
MMCALDS325	2.0	2.75	2.5	1.0	2.5	0.0	1.25	1.75	1.25	2.5	**1.75**
MMCALWP325	2.00	3.00	2.8	1.0	2.5	1.3	1.0	2.5	1.0	2.5	**1.96**
MMCAPPI425	-	-	-	-	-	-	-	-	-	-	-
MMCAPDI425	-	-	-	-	-	-	-	-	-	-	-
MMCAPSD425	-	-	-	-	-	-	-	-	-	-	-
MMCAPRC425	-	-	-	-	-	-	-	-	-	-	-
AVERAGE PLO	**2.45**	**2.38**	**2.39**	**1.45**	**2.52**	**1.35**	**1.67**	**1.96**	**1.46**	**2.21**	**1.642**