DCE-IV

To be effective from year-2025

COURSE TITLE: Advanced Computer Networks								
Course Code: MMCADAC225						**Examination Scheme**	**T**	**P**
Total number of Lecture Hours:60						**External**	**72**	**-**
						Internal	**28**	**-**
Lecture (L):	**4**	**Practical's (P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**4**	

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.

To be effective from year-2025

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.

To be effective from year-2025

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	PLO5	PLO 6	PLO 7	PLO 8	PLO 9	PLO10	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

	PLOs										
Unit wise CLOs	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**

To be effective from year-2025

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.

To be effective from year-2025

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.**

To be effective from year-2025

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**

COURSE TITLE: Web Programming								
Course Code: MMCACWP325						**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 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	

To be effective from year-2025

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**

To be effective from year-2025

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**

To be effective from year-2025

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

To be effective from year-2025

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.**

To be effective from year-2025

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 manipulatio clearing logs, spoofing, Anonymity Tools: Proxy servers, VPNs, and Tor, Tools Proxy-Chains, Tor Browser, Reporting and Documentation: Importance of repo 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