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.

To be effective from year-2025

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):

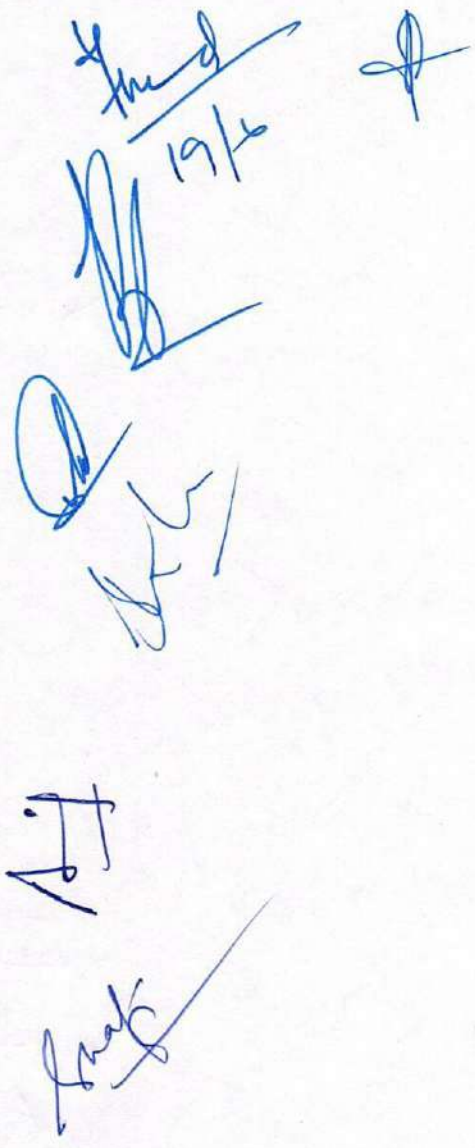
CLO1: Set up the Java development environment and apply fundamental programming constructs such as variables, loops, conditionals, arrays, and methods.

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



To be effective from year-2025

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

To be effective from year-2025

	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

To be effective from year-2025

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

To be effective from year-2025

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.

To be effective from year-2025

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

To be effective from year-2025

COURSE TITLE: Advanced Operating Systems

Course Code: MMCADA0225						**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 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**

To be effective from year-2025

COURSE TITLE: Digital Image Processing

Course Code: MMCADDI225						**Examination Scheme**	**T**	**P**	
Total number of Lecture Hours: 60						**External**	**72**	**-**	
Total number of Practical Hours: -						**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.	

To be effective from year-2025

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

To be effective from year-2025

COURSE TITLE: Decision Support Systems

Course Code: MMCADDS225						**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 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.	

To be effective from year-2025

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

	PLOs										
Unit-Wise CLOs	**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**

To be effective from year-2025

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.

To be effective from year-2025

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**