Kurukshetra University, Kurukshetra

(Established by the State Legislature Act-XII of 1956)
("A++" Grade, NAAC Accredited)

Syllabus
for

Post Graduate Programme

Master of Computer Applications
as per NEP-2020
Curriculum and Credit Framework for Postgraduate Programme

CBCS-LOCF
(3rd and 4th Semesters)
With effect from the session 2025-26

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
FACULTY OF SCIENCES

KURUKSHETRA UNIVERSITY, KURUKSHETRA -136119

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Design and Analysis of Algorithms		
Course Code	M24-CAP-301		
Course Type	CC-9		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course aims to provide students with a deep understanding of algorithm design, analysis, and complexity, equipping them with the skills to develop efficient solutions to computational problems. It introduces foundational concepts such as asymptotic notation and algorithmic paradigms like divide-and-conquer, greedy methods, and dynamic programming. Students will explore advanced data structures, graph and string algorithms, and will also gain exposure to probabilistic analysis and randomized algorithms. The course further delves into computational geometry and the theoretical aspects of computational complexity, including NP-completeness and intractable problems, preparing students for both practical algorithm development and theoretical analysis.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Students will be able to analyze and compare algorithm performance using asymptotic notations and apply divide-and-conquer strategies with recurrence-solving techniques. **CLO-2:** Students will develop efficient algorithmic solutions using dynamic programming and greedy methods for problems like matrix chain multiplication, Huffman coding, and optimal binary search trees. **CLO-3:** Students will understand and implement key graph and string algorithms, including shortest path algorithms, maximum flow techniques, and pattern matching methods. **CLO-4:** Students will comprehend geometric algorithmic techniques and critically evaluate problem complexity using concepts from computational complexity theory, including NP-completeness and reducibility.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction: Algorithms, Role of algorithms in computing, Analyzing algorithms, Designing algorithms, Asymptotic notations. Divide and Conquer: Solving recurrence equations: Back substitution method, Recursion tree method, Masters theorem. Probabilistic Analysis and Randomized Algorithms: The hiring problem, Indicator random variables, Randomized algorithms, Probabilistic analysis and further uses of indicator random variables	15

II	Trees: Red-black trees and Splay trees. Dynamic Programming (DP): Elements of DP, Matrix chain multiplication, Longest common subsequence, optimal binary search trees. Greedy Techniques (GT): Elements of GT, Activity selection problem, Huffman codes, Knapsack Problem.	15
III	Graph Algorithms: Topological sort, Strongly connected components, Single source shortest path: Analysis of Dijkstra's Algorithm, Limitations of Dijkstra's Algorithm, Negative weight cycle, Bellman-Ford algorithm. All Pairs Shortest Path: Relation of Shortest path and matrix multiplication, Analysis of Floyd Warshall algorithm. Maximum Flow: Flow network, Ford-Fulkerson method. Strings: Storage of strings, Naive string-matching algorithm, Rabin-Karp algorithm, String matching with finite automata, Knuth-Morris-Pratt algorithm	15
IV	Computational Geometry: Line-segment properties, Convex hull, Closest pair of points. Computational complexity: Notion of Polynomial time algorithms, Complexity classes: P, NP, NP-Hard and NP-Complete, Polynomial time verification, Reducibility, NP-Completeness, Examples of NP-Complete and NP-Hard problems: Traveling Salesman Problem, Knapsack, Bin Packing, Satisfiability, Vertex Cover, Clique, Independent Set.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) 1. Cormen, Leiserson, Rivest, Introduction to Algorithms, PHI India.
2) Neapolitan R., Foundations of Algorithms, Jones and Bartlett Learning.
3) Cooper A., Computability Theory, Chapman and Hall/ CRC Press.
4) A.V.Aho, J.E.Hopcroft, and J.D.Ullman, The Design and Analysis of Computer Algorithms, Pearson Education India
5) AnanyLevitin, Introduction to the Design and Analysis of Algorithms, Pearson Education.
6) R.C.T Lee, S.S.Tseng, R.C.Chang, Y.T.Tsai, Introduction to Design and Analysis of Algorithms: A Strategic Approach, Tata McGraw Hill
7) Steven Skiena, The Algorithm Design Manual, Springer India.

CC-10 Machine Learning in Python

With effect from the Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	3rd
Name of the Course	Machine Learning in Python
Course Code	M24-CAP-302
Course Type	CC-10
Level of the course (As per Annexure-I	500-599
Pre-requisite for the course (if any)	-
Course Objectives	The course Machine Learning in Python is designed to provide students with a comprehensive understanding of machine learning concepts and their practical implementation using Python. It introduces learners to the fundamentals of supervised and unsupervised learning, covering a range of algorithms including decision trees, naive Bayes, logistic regression, clustering methods, and regression models. The course also emphasizes hands-on experience through Python libraries like `scikit-learn`, `scipy`, and others to implement, train, and evaluate machine learning models. Furthermore, students will explore advanced topics such as dimensionality reduction, anomaly detection, recommender systems, support vector machines, and reinforcement learning, equipping them with both theoretical insight and practical tools to solve real-world problems using machine learning techniques.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO I**: Students will be able to explain fundamental machine learning concepts and implement classification tasks using decision tree algorithms in Python. **CLO II**: Students will demonstrate the ability to apply Bayesian methods and regression techniques using Python to model and predict data-driven outcomes. **CLO III**: Students will implement and analyze logistic regression and clustering algorithms to extract patterns and perform classification in unlabelled datasets using Python libraries. **CLO IV**: Students will apply dimensionality reduction and anomaly detection techniques and gain exposure to advanced machine learning topics such as SVM, recommender systems, and reinforcement learning using Python tools.

	Theory	Practical	Total
Credits	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to Machine Learning in Python**: Concepts and definitions, Overview of Machine Learning, Key terminology, types of learning tasks, and real-world applications. **Supervised Learning with Python**: Classification tasks, Representation of Decision Trees, identifying suitable problems for decision learning, Decision Tree algorithms (e.g., ID3, CART), and implementation using Python libraries such as `scikit-learn`. Exploration of hyperspace search in decision tree models.	15
II	**Naive Bayes in Python**: Bayes Theorem, applying Bayes Decision Theory, Conditional	15

	Probability, and implementation of Bayesian classifiers. Introduction to Bayesian Belief Networks and their applications using Python libraries. **Regression Models with Python**: Linear Regression for numerical prediction, methods for finding the best fit line, implementation using `scikit-learn`. Introduction to Regression Trees and performing regression using CART in Python.	
III	**Logistic Regression with Python**: Binary classification using the sigmoid function, model interpretation, and implementation using Python. **Clustering Techniques in Python**: Unsupervised learning concepts, implementation of K-Means Clustering, Expectation-Maximization (EM) algorithm, and Hierarchical Clustering using `scipy` and `scikit-learn`. Introduction to semi-supervised learning and supervised learning post-clustering.	15
IV	**Dimensionality Reduction and Anomaly Detection in Python**: Overview of dimensionality reduction techniques, implementation of Principal Component Analysis (PCA), detecting outliers using statistical and model-based approaches. **Advanced Topics**: Introduction to Recommender Systems using collaborative filtering, Support Vector Machines (SVM) for classification, and basics of Reinforcement Learning with an overview of relevant Python tools like `OpenAI Gym`.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
• Class Participation:	5	Written Examination	
• Seminar/presentation/assignment/quiz/class test etc.:	10		
• Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Tom M. Mitchell, Machine Learning, McGraw-Hill Education (India) Private Limited.
2) Ethem Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press.
3) Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press.
4) Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press.
5) Peter Harrington, Machine Learning in Action, Manning Publications.
6) Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press.
7) Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd or 3rd Edition), O'Reilly Media.

DEC-1 Ethical Hacking

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Ethical Hacking		
Course Code	M24-CAP-303		
Course Type	DEC-1		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)			
Course Objectives	The objective of this course is to provide students with an in-depth understanding of the ethical aspects of hacking and cybersecurity practices. It aims to develop skills in identifying security vulnerabilities in computer systems and networks through ethical means. The course introduces tools and techniques used by ethical hackers to conduct penetration testing, vulnerability assessments, and security audits. It also emphasizes the legal and ethical boundaries of hacking and trains students to defend against cyber threats while complying with industry standards and regulations.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to explain the basic concepts of ethical hacking and perform information gathering through footprinting and scanning techniques within legal boundaries. **CLO 2**: Students will analyze system vulnerabilities and demonstrate the ability to detect and mitigate various forms of malware and unauthorized system access. **CLO 3**: Students will assess and exploit weaknesses in networks and web applications while understanding the use of tools and methods used by ethical hackers. **CLO 4**: Students will perform penetration testing using industry-standard tools, interpret vulnerability reports, and communicate findings through professional documentation.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Ethical Hacking: Definition and types of hackers, phases of hacking, footprinting and reconnaissance, scanning networks and enumeration, understanding vulnerabilities, legal and ethical issues in ethical hacking, cyber laws in India and internationally.	15
II	System Hacking and Malware Threats: Password cracking techniques, privilege escalation, steganography and hiding data, malware types and analysis (viruses, worms, trojans), sniffing and spoofing, keyloggers and spyware, identifying and mitigating malware threats.	15
III	Network and Web Application Hacking: Understanding firewalls and intrusion detection systems (IDS), hacking wired and wireless networks, SQL injection, cross-site scripting (XSS), session hijacking, denial-of-service (DoS) attacks, web server vulnerabilities, exploiting web application weaknesses using tools such as Burp Suite.	15
IV	Penetration Testing and Security Tools: Phases of penetration testing, vulnerability assessment tools (Nmap, Nessus), exploiting systems using Metasploit Framework, writing and customizing scripts for ethical hacking, report writing and documentation, recent trends in ethical hacking.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30	**End Term Examination: 70**

➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) William Stallings, Network Security Essentials: Applications and Standards, Pearson Education.
2) William Easttom, Computer Security Fundamentals, Pearson Education.
3) Patrick Engebretson, The Basics of Hacking and Penetration Testing, Syngress.
4) Michael T. Simpson, Kent Backman, and James Corley, Hands-On Ethical Hacking and Network Defense, Cengage Learning.
5) EC-Council, Certified Ethical Hacker (CEH) Official Study Guide, Wiley.

DEC-1 Computer Architecture

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Computer Architecture		
Course Code	M24-CAP-304		
Course Type	DEC-1		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	Student must have the knowledge of Computer Organization & Design		
Course Objectives	This course aims to introduce students to the foundational models and architectures of parallel computation, emphasizing how hardware and software interact in parallel processing environments. It explores various types and levels of parallelism, including instruction-level parallelism (ILP) and the architecture of superscalar processors. Students learn about code scheduling, branch prediction strategies, and memory models in MIMD systems. The course also delves into interconnection networks and cache coherence protocols, equipping learners with the conceptual and practical knowledge needed to understand, analyze, and optimize modern parallel computing systems.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Learn the concepts of computational models & parallel architectures and exploitation of parallelism at instruction level. **CLO-2** Understand architectural features of multi-issue ILP processors and branch handling. **CLO-3** Learn MIMD architectures and interconnection networks used in them and evaluate their comparative performances. **CLO-4** Analyze causes of cache coherence problem and learn protocols for its solution.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Computational Model:** Computational models, computer architecture, Classification of parallel architectures, Relationships between programming languages and parallel architectures. **Parallel Processing:** Types and levels of parallelism, Instruction Level Parallel (ILP) processors, dependencies between instructions, principle and general structure of pipelines, performance measures of pipeline, pipelined processing instructions. **Code Scheduling for ILP- Processors:** Basic block scheduling, loop scheduling, global scheduling.	15
II	**Superscalar Processors:** Emergence of superscalar processors, Tasks of superscalar processing – parallel decoding, superscalar instruction issue, shelving, register renaming, parallel execution, preserving sequential consistency of instruction execution and exception processing, comparison of VLIW & superscalar processors. **Branch Handling:** Branch problem, Approaches to branch handling – delayed branching, branch detection and prediction schemes, branch penalties and schemes to reduce them, multiway branches.	15
III	**MIMD Architectures:** Concepts of distributed and shared memory MIMD architectures, UMA, NUMA, CC-NUMA & COMA models of multiprocessors. **Direct Interconnection Networks:** Metrics for performance measures; **Interconnection topologies:** Linear Array, Ring, Chordal Rings, Star, Tree, 2D Mesh, Barrel Shifter, Hypercube.	15
IV	**Dynamic interconnection networks:** single shared buses, comparison of bandwidths of locked, pended & split transaction buses, arbiter logics, crossbar, multistage networks – omega. **Cache Coherence Protocols:** Cache Coherence Problem, Causes, Hardware Based Protocols – Snoopy	15

	Cache Protocol, Directory Schemes – Full-Map Directory, Limited Directory, Chained Directory.	
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) Sima, Fountain, Kacsuk, Advanced Computer Architecture, Pearson Education.
2) D. A. Patterson and J. L. Hennessey, Computer Architecture – A Quantitative Approach, Elsevier India.

Reference Books

1) Kai Hwang, Advanced Computer Architecture, McGraw Hill.
2) Nicholas Carter, Computer Architecture, McGraw Hill.
3) Harry F. Jordan, Gita Alaghband, Fundamentals of Parallel Processing, Pearson Education.

DEC-2 Computer Graphics

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Computer Graphics		
Course Code	M24-CAP-306		
Course Type	DEC-2		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)			
Course Objectives	The objective of this course is to introduce students to the fundamental concepts and techniques of computer graphics, including both two-dimensional and three-dimensional modeling and rendering. The course aims to develop an understanding of the components, devices, and software involved in computer graphics systems, and to explore the mathematical foundations of graphical transformations and geometric representations. Students will learn how to implement various algorithms for scan conversion, clipping, filling, and rendering of objects, along with basic animation techniques. The course also covers advanced topics such as 3D viewing, visible-surface determination, shading models, and procedural modeling methods to give students insight into realistic visualizations and modern applications of computer graphics such as GUI design, computer vision, and augmented reality.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to describe the components and working principles of interactive computer graphics systems and distinguish between raster and random scan display mechanisms. **CLO 2**: Students will demonstrate the ability to implement and analyze scan conversion algorithms for drawing basic geometric shapes and understand curve representation techniques using parametric equations and Bézier curves. **CLO 3**: Students will apply two-dimensional geometric transformations and viewing techniques, implement clipping algorithms, and effectively perform operations such as zooming, panning, and window-to-viewport transformations. **CLO 4**: Students will gain proficiency in 3D object modeling, transformations, projections, surface visibility determination techniques, and will understand the fundamentals of illumination, shading, and animation sequence design.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		
Part B- Contents of the Course			

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Computer Graphics and its Components: Overview of Computer Graphics, its functions & elements; Introduction to GUI, Computer Vision, Augmented Reality and other Applications of Graphics; Popular Graphics Software; Components and Working of Interactive Graphics; Raster Scan and Random Scan systems and Display Processors; Look-up table; Loading the Frame Buffer; Coordinate Systems. Graphics Devices: Display Technologies: Resolution, Aspect Ratio, Refresh CRT, Color CRT, Flat Panel Displays; Interactive Input Devices for Graphics, Image and Video Input Devices.	15
II	Scan Conversion: Drawing Geometry; Output Primitives; Lines and Pixel Graphics; AntiAliasing; Scan Converting Lines: DDA line drawing algorithms, Bresenham's line Algorithm; Scan Converting Circles: Polynomial method for circle drawing, circle drawing using polar coordinates, Bresenham's circle drawing; Algorithms for Generation of ellipse; Line Styles; Generation of Bar Charts, Pie-Charts. Curve Representation: Parametric Curves, Parametric Representation of a Circle, Parametric representation of cubic curves, drawing Bezier curves.	15

	Filled-Area Primitives: Basic Stack based fill algorithms: Flood fill algorithm, Boundary fill algorithm; Scan-line polygon fill algorithm and its computational structures.	
III	Two-Dimensional Transformations: Coordinate and Geometric Transformations; Translation, Rotation, Scaling; Matrix representations and Homogeneous coordinates, Composite transformations, General Pivot Point rotation, General Fixed Point Scaling, Shearing; Reflection ; Reflection about an arbitrary line. 2-D Viewing: Viewing pipeline; Window, Viewport, Window-to-Viewport transformation; Zooming, Panning; Pointing and Positioning techniques; Rubber band technique; Dragging. Clipping operations: Point and Line clipping, Cohen-Sutherland line clipping, Mid-Point Subdivision line clipping, Liang-Barsky line clipping, Sutherland-Hodgman polygon clipping; Weiler-Atherton polygon clipping.	15
IV	3-D Graphics & Modeling: Visualization techniques for Realism; 3D Object Representation; Solid Model Representation Schemes; Euclidean Geometry methods: Regularized Boolean Set Operations, Primitive Instancing, Boundary Representations, Curved lines and surfaces, Sweep Representations, Spatial-Partitioning Representations - Octree representation, Constructive Solid Geometry; Procedural Methods: Fractals, Shape Grammars, Particle systems, Physically Based modeling, Visualization techniques; 3D transformations. Visible-Surface Determination: Z-buffer, Depth-Sorting, Area Subdivision, BSP-Tree method; Ray casting. Illumination and Shading: Modeling Light Intensities; Basic Illumination Models; Gouraud Shading; Phong Shading.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Donald Hearn, M. Pauline Baker, Computer Graphics, Pearson Education.
2) J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics - Principles and Practice, Pearson Education.
3) Newmann & Sproull, Principles of Interactive Computer Graphics, McGraw Hill.
4) Rogers, David F., Procedural Elements of Computer Graphics, McGraw Hill.
5) Zhigang Xiang, Roy Plastock, Computer Graphics, Tata McGraw Hill.
6) Malay K. Pakhira, Computer Graphics, Multimedia and Animation, PHI
7) Steven Harrington, Computer Graphics, A Programming Approach, McGraw Hill.

DEC-2 Big Data & Pattern Recognition

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Big Data & Pattern Recognition		
Course Code	M24-CAP-307		
Course Type	DEC-2		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)			
Course Objectives	The objective of this course is to provide students with a comprehensive understanding of the core concepts, technologies, and analytical methods used in the domain of big data and pattern recognition. The course introduces the foundational principles of big data, including its characteristics, sources, business motivations, and the lifecycle of big data analytics. Students will learn about various enterprise technologies and data governance strategies necessary for managing and analyzing large-scale datasets effectively. The course further explores the tools and techniques used for data processing, such as Hadoop ecosystem components, and offers insight into the theoretical and practical aspects of pattern recognition, including feature extraction, classification, and neural networks. It concludes with an overview of NoSQL databases, highlighting their architectures, types, and applications in big data management, enabling students to distinguish between traditional and modern data handling approaches.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to explain the fundamental characteristics of big data and identify its business drivers, architectures, and real-world applications across various industries. **CLO 2**: Students will demonstrate the ability to apply big data governance principles and utilize core Hadoop ecosystem tools such as HDFS, MapReduce, Hive, and Pig for analyzing large-scale datasets. **CLO 3**: Students will apply pattern recognition techniques, including feature extraction and nearest neighbor classifiers, and model simple logical operations using neural networks. **CLO 4**: Students will analyze the characteristics and types of NoSQL databases and evaluate their suitability for different big data storage and retrieval scenarios compared to traditional RDBMS.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		
Part B- Contents of the Course			

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Understanding Big Data: Concepts and Terminology, Big Data Characteristics, Different Types of Data, Identifying Data Characteristics, Business Motivations and Drivers for Big Data Adoption: Business Architecture, Business Process Management, Information and Communication Technology, Big Data Analytics Lifecycle, Enterprise Technologies and Big Data Business Intelligence, Industry examples of big data.	15
II	Data Governance for Big Data Analytics: Evolution of Data Governance, Big Data and Data Governance, Big Datasets, Big Data Oversight, Big Data Tools and Techniques: HDFS, Map Reduce, YARN, Zookeeper, HBase, HIVE, Pig, Mahout, Developing Big Data Applications, Stepwise Approach to Big Data Analysis, Big Data Failure: Failure is common, Failed Standards, Legalities.	15
III	Data Analysis and Pattern Recognition: Quantitative and Qualitative Analysis, Pattern Recognition	15

	Systems, Fundamental Problems in Pattern Recognition, Feature Extraction and Reduction, Paradigms, Pattern Recognition Approaches, Importance and Applications. Data Domain for Pattern Recognition. Pattern Recognition using Nearest Neighbour Classifier and Modeling an AND Gate Neural Nets.	
IV	An Overview of NoSQL, Characteristics of NoSQL, NoSQL Storage Types, Introduction of NoSQL Products, NoSQL Data Management for Big Data: Schema Less Models, Key-Value Stores, Document Stores, Tabular Stores, Object Data Stores, Graph databases, NoSQL Misconceptions, NoSQL over RDBMS.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Thomas Erl, Wajid Khattak and Paul Buhler, Big Data Fundamentals Concepts, Drivers & Techniques Prentice Hall.
2) David Loshin, Big Data Analytics from Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph Morgan Kaufmann.
3) Jules J. Berman, Principles of Big Data Preparing, Sharing and Analyzing Complex Information, Morgan Kaufmann.
4) Gaurav Vaish, Getting Started with NoSQL, Packt Publishing.
5) Rajjan Shinghal, Pattern Recognition Techniques and Applications, Oxford Higher Education.
6) Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer.
7) Jay Liebowitz, Big Data and Business Analytics, Auerbach Publications, CRC press.
8) Pete Warden, Big Data Glossary, O'Reily.
9) Michael Mineli, Michele Chambers, AmbigaDhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley Publications.

With effect from Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	3rd		
Name of the Course	Practical-7		
Course Code	M24-CAP-309		
Course Type	PC-5		
Level of the course	500-599		
Pre-requisite for the course (if any)			
Course objectives	This course aims to equip students with hands-on experience in implementing and analyzing fundamental and advanced algorithms. Students will develop the ability to measure and compare the empirical time complexity of sorting algorithms, trace recursion trees, and understand the behavior of divide-and-conquer strategies. They will gain practical skills in implementing dynamic programming solutions for problems such as Matrix Chain Multiplication and Longest Common Subsequence, and in applying advanced data structures like Red-Black Trees. The course also emphasizes solving real-world problems using greedy, graph-based, and optimization algorithms such as Huffman Coding, 0/1 Knapsack, Bellman-Ford, and Floyd-Warshall. By the end of the course, students will be proficient in algorithmic problem-solving and performance analysis, preparing them for software development, competitive programming, and research.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Implement and empirically evaluate the performance of classical algorithms such as Bubble Sort, Merge Sort, and Quick Sort, and analyze their execution times across different input cases. **CLO-2:** Apply and trace advanced algorithmic techniques, including recursion trees in divide-and-conquer approaches and dynamic programming tables in optimization problems like Matrix Chain Multiplication and LCS. **CLO-3:** Design and implement efficient data structures and graph algorithms, such as Red-Black Trees, Huffman Coding, Topological Sort, and algorithms for shortest paths and strongly connected components. **CLO-4:** Compare and contrast greedy and dynamic programming approaches for problems like the 0/1 Knapsack and understand their suitability through programmatic implementations and results analysis.		
Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course	
Practicals	**Contact Hours**
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 3 questions from the Part-A and 2 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and one from the Part-B.	120
Part-A 1) Write a program to compare the execution time of different sorting algorithms (Bubble Sort, Merge Sort, Quick Sort) and analyze their empirical time complexity. 2) Implement the Merge Sort. Trace the recursion tree and compare the number of recursive calls for different inputs. 3) Implement the Quick Sort algorithms. Trace the recursion tree and compare the number of recursive calls for different inputs. 4) Apply Strassen's Matrix Multiplication and compare with the naive matrix multiplication algorithm.	60

5) Implement Red-Black Tree with insertion and deletion. Validate the tree's properties at each step. 6) Solve the Matrix Chain Multiplication problem using dynamic programming. 7) Write a program to find the Longest Common Subsequence of two strings and trace the DP table.	
Part-B 8) Construct Huffman Codes for a given set of characters and frequencies; also display the encoded and decoded message. 9) Solve the 0/1 Knapsack Problem using Greedy approache. 10) Solve the 0/1 Knapsack Problem using DP approaches. 11) Implement Topological Sort using DFS algorithm. 12) Implement Kosaraju's algorithm to find strongly connected components in a directed graph. 13) Implement Bellman-Ford algorithm to find shortest paths and detect negative weight cycles. 14) Implement the Floyd-Warshall algorithm for all-pairs shortest paths.	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Cormen, Leiserson, Rivest, Introduction to Algorithms, PHI India.
2) Neapolitan R., Foundations of Algorithms, Jones and Bartlett Learning.
3) Cooper A., Computability Theory, Chapman and Hall/ CRC Press.
4) A.V.Aho, J.E.Hopcroft, and J.D.Ullman, The Design and Analysis of Computer Algorithms, Pearson Education India
5) AnanyLevitin, Introduction to the Design and Analysis of Algorithms, Pearson Education.
6) R.C.T Lee, S.S.Tseng, R.C.Chang, Y.T.Tsai, Introduction to Design and Analysis of Algorithms: A Strategic Approach, Tata McGraw Hill
7) Steven Skiena, The Algorithm Design Manual, Springer India.

PC-6 Practical-8

With effect from Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	3rd
Name of the Course	Practical-8
Course Code	M24-CAP-310
Course Type	PC-6
Level of the course	500-599
Pre-requisite for the course (if any)	
Course objectives	This course aims to equip students with practical experience in implementing foundational machine learning algorithms using Python. It covers key supervised learning techniques such as decision trees, naive Bayes, linear and logistic regression, and introduces students to unsupervised learning techniques like clustering and dimensionality reduction. Through hands-on projects, students will explore model evaluation metrics, visualize model behaviors, and analyze datasets using libraries like `scikit-learn`, `matplotlib`, and `pandas`. The course also integrates basic anomaly detection approaches, enabling students to identify unusual patterns in data. Overall, the course bridges theoretical concepts with applied programming, preparing students to solve real-world problems using machine learning tools.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Apply supervised learning techniques such as decision trees, naive Bayes, and linear/logistic regression to solve classification and prediction problems using Python. **CLO-2:** Perform model evaluation using metrics like R^2 score, mean squared error, and visualize model behavior through regression lines and decision boundaries. **CLO-3:** Implement unsupervised learning techniques, including K-Means clustering and Principal Component Analysis (PCA), and interpret results through appropriate visualizations. **CLO-4:** Detect anomalies in datasets using statistical and model-based methods like Z-score, Isolation Forest, and One-Class SVM for applications such as fraud detection and system monitoring.

	Theory	Practical	Total
Credits	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	Contact Hours
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 3 questions from the Part-A and 2 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and one from the Part-B.	120
Part-A 1) Implement a Decision Tree Classifier on the Iris dataset using scikit-learn and visualize the decision tree. 2) Implement a Gaussian Naive Bayes Classifier on the Wine dataset using `scikit-learn`. 3) Implement Simple Linear Regression using `scikit-learn` to predict house prices. 4) Implement Multiple Linear Regression and evaluate R^2 score, MSE, and other metrics on a real-world dataset. 5) Write a program to plot the regression line along with the data points using `matplotlib`.	60

6) Perform regression using CART (Decision Tree Regressor) and compare performance with linear regression.	
Part-B 7) Apply Logistic Regression for binary classification on a heart disease dataset using `scikit-learn`. 8) Visualize the sigmoid function and decision boundary of logistic regression on a binary classification dataset. 9) Implement K-Means Clustering on a synthetic dataset and visualize the clusters. 10) Apply K-Means on customer segmentation data, and determine the optimal number of clusters using the Elbow Method. 11) Implement PCA on a high-dimensional dataset and visualize the first two principal components. 12) Create a program for outlier detection using the Z-score method on a numeric dataset. 13) Use Isolation Forest or One-Class SVM to detect anomalies in network traffic or financial transactions.	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Practicum**	**30**	➢ **Practicum**	**70**
• Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
• Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
• Mid-Term Examination:	15		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Tom M. Mitchell, Machine Learning, McGraw-Hill Education (India) Private Limited.
2) Ethem Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press.
3) Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press.
4) Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press.
5) Peter Harrington, Machine Learning in Action, Manning Publications.
6) Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press.
7) Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd or 3rd Edition), O'Reilly Media.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of Programme	MCA		
Semester	3rd		
Name of the Course	Data Analytics using Excel		
Course Code	M24-OEC-308		
Course Type	OEC		
Level of the course	500-599		
Pre-requisite for the course (if an)			
Course Objectives	The objective of this course is to equip students with practical knowledge and hands-on skills in performing data analysis using Microsoft Excel. The course introduces the fundamentals of data analytics, focusing on the importance of structured data processing and interpretation for decision-making. Students will learn to work with Excel functions, data formatting, and visualization tools to perform descriptive analytics. The course also covers advanced Excel features such as PivotTables, What-if analysis, data models, and statistical functions for predictive analysis. By the end of the course, students will be capable of cleaning, analyzing, visualizing, and interpreting datasets using Excel, supporting business and research-oriented data-driven decisions.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to understand core concepts of data analytics and effectively use Excel to organize, format, and visualize raw data. **CLO 2**: Students will demonstrate the ability to perform descriptive statistical analysis and apply data-cleaning techniques using Excel's built-in features. **CLO 3**: Students will be able to apply advanced Excel tools like PivotTables and What-if analysis to derive insights and perform basic forecasting. **CLO 4**: Students will conduct statistical and predictive analysis using Excel's built-in tools and interpret the results through dynamic dashboards and visual reports.		
Credits	Theory	Tutorial	Total
	2		2
Teaching Hours per week	2		2
Internal Assessment Marks	15		15
End Term Exam Marks	35		35
Max. Marks	50		50
Examination Time	3		3

Part B- Contents of the Course

<u>Instructions for Paper- Setter:</u> The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the
compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Introduction to Data Analytics: Definition, types of analytics (descriptive, predictive, prescriptive), data types and sources, role of Excel in analytics. Excel basics: interface, cell references, formulas, functions, sorting and filtering data, data formatting, basic charts and graphs, importing data from different sources.	8
II	Descriptive Statistics using Excel: Calculating mean, median, mode, standard deviation, variance, percentiles, quartiles, and correlation. Using Excel functions for data summarization and exploration. Creating frequency distributions and histograms. Data cleaning and transformation using Text-to-Columns, Flash Fill, and Conditional Formatting.	7
III	Data Analysis Tools in Excel: Using PivotTables and PivotCharts for summarization and trend analysis. Introduction to data models and Power Pivot. Performing What-if Analysis using Goal Seek, Scenario Manager, and Data Tables. Introduction to forecasting using Excel functions like TREND and	8

	LINEST.	
IV	Statistical and Predictive Analysis: Performing regression analysis in Excel. Using correlation, ANOVA, and t-tests. Introduction to the Data Analysis Toolpak for hypothesis testing. Creating dashboards with interactive elements such as slicers and form controls. Case studies involving real-world datasets to interpret and communicate analytical results effectively.	7
Total Contact Hour		**30**

Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Theory**	**15**	➢ **Theory:**	**35**
➢ Class Participation:	4	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	4		
➢ Mid-Term Exam:	7		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Wayne Winston, Microsoft Excel Data Analysis and Business Modeling, Microsoft Press.
2) Paul McFedries, Excel Data Analysis for Dummies, Wiley.
3) Danielle Stein Fairhurst, Using Excel for Business Analysis, Wiley.
4) Conrad Carlberg, Statistical Analysis: Microsoft Excel 2016, Que Publishing.
5) M. Balasubramanian, Data Analysis Using Excel, McGraw Hill Education.

DEC-3 Object Oriented Design and UmL

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4th		
Name of the Course	Object Oriented Design and UmL		
Course Code	M24-CAP-401		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course provides a comprehensive understanding of Unified Modeling Language (UML) and object-oriented design principles for effective software development. It introduces the conceptual model of UML, including its various elements like use cases, class diagrams, state diagrams, and interaction diagrams. Students learn to gather and analyze software requirements, model system structure and behavior, and use UML diagrams to design scalable and maintainable systems. The course emphasizes both high-level architectural planning and detailed class and object modeling, equipping students with the skills needed to transition from requirements to implementation effectively.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Understands basics of modelling and UML and applying usecase modeling for requirements elicitation **CLO-2** Understand and apply concepts of class modeling and advanced structural modeling through UML diagrams. **CLO-3** Understand and apply behavioural modeling through state diagram, sequence diagram, activity diagram to represent software systems. **CLO-4** Have a working ability and grasping attitude to analyse and design software systems based on object-oriented thinking using UML.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

<u>Instructions for Paper- Setter:</u> The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Introduction to UML:** Principles of Modeling, Conceptual Model of UML - Things, Relationships, Diagrams; Mechanisms in The UML – Specifications, Adornments, Common Divisions, Extensibility Mechanisms. **Usecase Modeling:** Identifying Use Cases & Actors, Relationships - between Actors, between Use Cases and between Actor and Use Case, Elements of Use Case Diagram, Constructing Use Case Diagram. **Requirement Engineering:** Gathering and Documenting Requirements, Functional Vs. Non-Functional Requirements, Requirement Validation and Verification.	15
II	**Structural Modeling with UML:** Class Modeling - Generalization & Inheritance, Association, Multiplicity, Sets, Bags & Sequences, Role Names, Qualified Association, Aggregation, Composition, Association Attributes, Association Classes, Abstract Class, Metadata, Reification, Elements of Class Diagrams, Constructing Class Diagrams. **Advanced Structural Modeling with UML:** Object Diagrams - Instances and Links; Package Diagrams - Organizing Classes, Components Diagram, Deployment Diagram.	15
III	**Behavioral Modeling with UML:** State Modeling - Events, States, Transitions & Conditions, Activity Effects, Do-Activities, Entry & Exit Activities, Completion Transitions, Elements of State Diagrams, Nested State Diagrams, Concurrency, Constructing State Diagrams. **Sequence Diagram:** Lifelines, Messages, and Activations, Constructing Sequence Diagram and Communication Diagram	15

	Activity Diagram: Actions, Control Flows, and Object Flows, Swim Lanes, Constructing Activity Diagram	
IV	**System Design:** Estimating Performance, Make A Reuse Plan, Organize The System into Subsystem, Identifying Concurrency, Allocating Subsystems to Processors and Tasks, Management of Data Stores, Handling Global Resources, Choosing Software Control Strategies, Handling Boundary Conditions, Setting Trade-Off Priorities, Selecting an Architect Style. **Class Design:** bridging gap, realize use cases with operations, designing algorithms, design optimization, adjustment of inheritance.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Pearson education.
2) M. Blaha, J. Rumbaugh, Object-Oriented Modeling and Design with UML, Pearson Education.

Reference Books

1) Brend Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering, Using UML, Patterns and Java, Pearson Education.
2) J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall of India.
3) Satzinger, Jackson, Burd, Object-Oriented Analysis & Design with the Unified Process, Thomson.
4) Grady Booch, Object Oriented Analysis & Design, Pearson Education.
5) Craig Larman , Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4th		
Name of the Course	Data Science		
Course Code	M24-CAP-402		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to introduce students to the foundational concepts and practical techniques used in data science. It aims to equip learners with the skills to collect, clean, analyze, and visualize data to generate meaningful insights. The course covers the end-to-end data science lifecycle, starting from data acquisition and preprocessing to exploratory data analysis, statistical modeling, and machine learning. Students will explore real-world data applications, understand the use of Python-based tools, and learn how to effectively communicate findings through visualization and storytelling. The course also emphasizes ethical considerations and best practices in handling data responsibly in decision-making contexts.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to explain the data science lifecycle and identify the roles and tools involved in solving real-world data problems. **CLO 2**: Students will demonstrate the ability to preprocess, clean, and explore datasets using Python-based libraries for meaningful data interpretation. **CLO 3**: Students will apply basic statistical and machine learning techniques to build and evaluate predictive models on structured data. **CLO 4**: Students will create effective data visualizations, interpret model outputs, and understand ethical responsibilities in data-driven decision-making.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Data Science: Definition and components of data science, data science vs. data analytics vs. machine learning, applications of data science in various domains. Data science lifecycle: data collection, data wrangling, analysis, interpretation, and communication. Overview of tools: Python, Jupyter, NumPy, pandas, and Matplotlib. Introduction to types of data and sources.	15
II	Data Preprocessing and Exploration: Handling missing values, data cleaning and transformation, outlier detection and treatment, encoding categorical variables, data normalization and standardization. Exploratory data analysis (EDA): using pandas and visualization libraries for summarizing and exploring datasets, correlation analysis, feature engineering.	15
III	Statistical Foundations and Machine Learning: Descriptive and inferential statistics, probability distributions, hypothesis testing, regression analysis. Introduction to supervised learning: linear regression, logistic regression, decision trees. Basics of unsupervised learning: clustering and dimensionality reduction. Model evaluation metrics: accuracy, precision, recall, F1-score, confusion matrix, cross-validation.	15
IV	Data Visualization and Communication: Principles of effective data visualization, tools for visualization in Python: Matplotlib, Seaborn, and Plotly. Dashboards and storytelling with data. Introduction to real-world case studies and capstone mini-projects. Ethical concerns in data science: data privacy, algorithmic bias, and responsible AI.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Joel Grus, Data Science from Scratch: First Principles with Python, O'Reilly Media.
2) Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly Media.
3) Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
4) Wes McKinney, Python for Data Analysis, O'Reilly Media.
5) Hadley Wickham and Garrett Grolemund, R for Data Science, O'Reilly Media (for supplementary reading and comparative learning).

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4^{th}		
Name of the Course	Theory of Computation		
Course Code	M24-CAP-403		
Course Type	DEC-3		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course introduces the foundational concepts of computation through the study of formal languages and automata theory. It covers finite automata (DFA, NFA, ε-NFA), regular expressions, and regular grammars, along with their equivalence and minimization techniques. Students learn about context-free grammars (CFG), derivation trees, grammar simplification, and pushdown automata (PDA) for recognizing CFLs. The course also explores finite automata with output (Moore and Mealy machines), Turing machines as general models of computation, and undecidable problems. By the end, learners will gain a solid understanding of the theoretical models that underpin programming languages and compiler design.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Design deterministic and non-deterministic finite state machines and understand their capabilities and limits **CLO-2** Define and discuss the concept of formal grammar, formal language, regular expression, and automata machine. **CLO-3** Design finite automata and push down automata machines for given formal languages or computational real-world problem statements. **CLO-4** Understand basic properties of Turing machines and computing with Turing machines.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Fundamentals:** Alphabets, Strings, Languages, Problems, Graphs, Trees, Finite State Systems, Definitions, Finite Automaton Model, Acceptance of Strings, And Languages. **Types of FA:** Deterministic Finite Automaton and Nondeterministic Finite Automaton, Transition Diagrams, Transition Tables, Equivalence of DFA and NFA. Finite Automata with □-Moves, Significance, Acceptance of Languages, □-Closure, Equivalence of NFA With and Without □-Moves, Minimization of Finite Automata, Myhill-Nerode Theorem.	15
II	**Finite Automata with Output**: Moore and Mealy Machines, Equivalence of Moore, and Mealy machines. **Regular Languages:** Regular Sets, Regular Expressions, Identity Rules, Constructing Finite Automata for A Given Regular Expressions, Conversion of Finite Automata to Regular Expressions. Arden Theorem, Pumping Lemma for Regular Languages, Closure Properties of Regular Sets, Regular Grammars–Right Linear and Left Linear Grammars.	15
III	**Context Free Grammar:** Context Free Grammar, Derivation Trees, Sentential Forms, Right Most and Leftmost derivation of Strings, Ambiguity in Grammar, Inherent ambiguity. Simplification of Context Free Grammars, Chomsky Normal Form, Greibach Normal Form, Pumping Lemma for Context Free Languages and Its Applications, Closure properties of CFL, Membership Algorithm (CYK Algorithm) for Context Free Grammars.	15
IV	**Push Down Automata (PDA):** Description and definition, Instantaneous Description, Language of PDA,	15

	Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, two stack PDA. **Turing machines (TM):** Linear Bounded Automata (LBA), Turing Machines (TM), General Model of Computation, Combining TM, Multi-Tape TM, Restricted and Universal TM, Recursive and recursively-enumerable languages and Properties, Reductions and the Halting Problem, Post's correspondence problem.	
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) J. E. Hopcroft, Rajeev Motwani and J. D. Ullman, "Introduction to Automata Theory, Languages, and Computation", 3rd Edition, Pearson Publisher, 2008, ISBN: 978-8131720479.

Reference Books

1) John C. Martin, "Introduction to Languages and the Theory of Computation", 4th Edition, McGraw Hill Education, 2010, ISBN: 978-0073191461.
2) K. L. P. Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata, languages and Computation", 3rd Edition, 2006, Prentice Hall of India, ISBN: 978-8120329683.
3) Peter Linz, "An Introduction to Formal Languages and Automata", 6th Edition, Jones & Bartlett Publisher, 2016, ISBN: 978-9384323219.
4) Zvi Kohav, Niraj K. Jha, "Switching and Finite Automata Theory", 3rd Edition, Cambridge University Press, 2009, ISBN: 978-0521857482..

Rakesh

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4th		
Name of the Course	Blockchain Technology		
Course Code	M24-CAP-405		
Course Type	DEC-4		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	The objective of this course is to provide students with a comprehensive understanding of blockchain technology, its architecture, core components, and potential applications in various domains. The course aims to introduce the theoretical foundations and the underlying cryptographic principles of blockchain while equipping students with hands-on knowledge of how decentralized systems work. Students will explore consensus mechanisms, smart contracts, and distributed ledger technologies. The course also emphasizes real-world use cases such as cryptocurrencies, supply chain management, identity verification, and decentralized finance. Through this course, learners will develop the necessary skills to analyze, design, and implement blockchain-based solutions securely and efficiently.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to explain the fundamental concepts and architecture of blockchain technology and compare it with traditional centralized systems. **CLO 2**: Students will understand the cryptographic techniques and consensus algorithms that ensure the security and integrity of blockchain networks. **CLO 3**: Students will demonstrate the ability to develop basic smart contracts using Solidity and understand the differences between blockchain platforms such as Ethereum and Hyperledger. **CLO 4**: Students will analyze the practical applications of blockchain technology, evaluate associated security challenges, and discuss emerging trends and future directions.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Blockchain: History and evolution of blockchain, features and benefits, architecture of blockchain, types of blockchain (public, private, consortium), distributed ledger technology, blockchain versus traditional databases, overview of blockchain applications across industries.	15
II	Cryptographic Foundations and Consensus Mechanisms: Hash functions, digital signatures, public-key cryptography, Merkle trees, proof-of-work (PoW), proof-of-stake (PoS), delegated proof-of-stake (DPoS), Byzantine fault tolerance, mining and validation processes, block structure, and block propagation.	15
III	Smart Contracts and Platforms: Concept of smart contracts, Ethereum blockchain and Ethereum Virtual Machine (EVM), Solidity programming basics, gas usage, writing and deploying simple smart contracts, introduction to Hyperledger Fabric and its components, comparison between Ethereum and Hyperledger.	15
IV	Blockchain Use Cases, Security, and Challenges: Blockchain in cryptocurrency (Bitcoin), finance, supply chain, healthcare, identity management, voting, and IoT. Security issues in blockchain including 51% attacks, double spending, Sybil attacks, scalability and interoperability challenges. Overview of recent advancements and future trends in blockchain.	15
	Total Contact Hours	60

<table>
<tr><th colspan="4">Suggested Evaluation Methods</th></tr>
<tr><th colspan="2">Internal Assessment: 30</th><th colspan="2">End Term Examination: 70</th></tr>
<tr><td>➢ Theory</td><td>30</td><td>➢ Theory</td><td>70</td></tr>
<tr><td>➢ Class Participation:</td><td>5</td><td colspan="2" rowspan="3">Written Examination</td></tr>
<tr><td>➢ Seminar/presentation/assignment/quiz/class test etc.:</td><td>10</td></tr>
<tr><td>➢ Mid-Term Exam:</td><td>15</td></tr>
<tr><th colspan="4">Part C-Learning Resources</th></tr>
</table>

Reference Books

1) Imran Bashir, Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications, Packt Publishing.
2) Andreas M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O’Reilly Media.
3) Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media.
4) Arshdeep Bahga and Vijay Madisetti, Blockchain Applications: A Hands-On Approach, VPT.
5) Joseph J. Bambara and Paul R. Allen, Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions, McGraw-Hill Education.

DEC-4 Biometric Security

With effect from the Session: 2024-25	
Part A - Introduction	
Name of the Programme	MCA
Semester	4th
Name of the Course	Biometric Security
Course Code	M24-CAP-406
Course Type	DEC-4
Level of the course (As per Annexure-I	500-599
Pre-requisite for the course (if any)	
Course Objectives	This course introduces the fundamentals and practical applications of biometric technologies for secure identification and authentication. It covers key biometric modalities (physiological and behavioral), system processes (enrollment, verification, identification), and performance metrics. Students will explore multimodal systems, biometric security challenges (e.g., spoofing, liveness detection), and integration strategies. The course also addresses ethical, privacy, and interoperability issues, with real-world applications and an introduction to soft and DNA biometrics.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1.** Understand the basics of biometric and its functionalities **CLO-2** Understand and analyze biometric systems with different traits. **CLO-3** Analyze the performance of biometrics by combining multiple traits and biometric sensors. **CLO-4** Understand various Biometric security issues & Applications

	Theory	Practical	Total
Credits	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

<u>Instructions for Paper- Setter:</u> The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Biometric; Biometric technologies, Biometric Enrollment, Identification & Verification process , Biometric advantages, Different Biometric traits, characteristics of ideal biometric, Comparison of different biometrics, Accuracy in biometric systems; Error rates, Biometric deformations.	15
II	Physiological and Behavioral Biometric; Different Physiological and Behavioral Biometrics, Fingerprint, Face, Iris Introduction, Recognition, Feature extraction and matching, IRIS segmentation- normalization, Gait feature extraction, Hand geometry; Behavioral Biometrics: Introduction, Features, classification and properties of behavioral biometrics; Signature, Keystroke dynamics, Voice etc., merits and their demerits, soft biometrics.	15
III	Multimodal Biometrics and Biometric Sensors; Multimodal and Multi-biometric systems, Biometric Integration strategies, Architecture, Level of fusion, Training and adaptability, examples of multimodal biometric systems, Performance evaluation. Different types of Biometric sensors and their use; Biometric sensor interoperability.	15
IV	Biometrics Security and Applications; Biometrics Security; Biometric system challenges, Attack on biometric system, Liveness detection, Cancelable biometrics. Biometric standards, Applications of biometrics; Government sector, Commercial sector, Privacy in biometrics, Biometric Ethics and Technology usage, DNA Biometrics.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		

➢ Mid-Term Exam:	15	

Part C-Learning Resources

Reference Books:

1. Anil K. Jain, Arun Ross, and Karthik Nandakumar, “Introduction to Biometrics”, Springer.
2. Anil K Jain, Patrick Flynn and Arun A Ross, “Handbook of Biometrics”, Springer.
3. James Wayman, Anil Jain, Davide Maltoni, Dario Maio, “Biometric Systems, Technology Design and Performance Evaluation”,Springer.
4. S.Y. Kung, S.H. Lin, M.W.Mak, “Biometric Authentication: A Machine Learning Approach” Prentice Hall.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4^{th}		
Name of the Course	Soft Computing		
Course Code	M24-CAP-407		
Course Type	DEC-4		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course provides a comprehensive introduction to soft computing techniques, including fuzzy logic, neural networks, and evolutionary algorithms. It covers metaheuristic optimization methods such as PSO, ACO, and other bio-inspired algorithms. Students will learn fuzzy set theory, inference systems, and design of fuzzy controllers. The course also explores various neural network models and genetic algorithms for solving real-world optimization and classification problems. Emphasis is placed on applications, limitations, and emerging trends in soft computing.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1:** Soft Computing and Optimization: Students will understand soft computing techniques and apply metaheuristic optimization methods to solve complex problems. **CLO-2:** Fuzzy Systems: Students will analyze fuzzy logic principles and implement fuzzy inference systems for decision-making. **CLO-3:** Neural Networks: Students will develop and apply neural network models for pattern recognition and machine learning. **CLO-4:** Genetic Algorithms: Students will design genetic algorithms and utilize evolutionary principles for optimization.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Soft Computing:** Conventional AI to Computational Intelligence; Soft Computing Constituents and Applications. **Introduction to Non-traditional Metaheuristic Optimization Techniques:** Random Optimization, Simulated Annealing, Tabu Search, Ant Colony Optimization, Particle Swarm Optimization, Harmony Search, Memetic Algorithms, Other Evolutionary Algorithms such as Firefly Algorithm, Bee Algorithm, Shuffled Frog Leap algorithm, Bat algorithm etc.	15
II	**Fuzzy Set theory:** Fuzzy Sets & Classical Sets; Operations on Fuzzy Sets, Fuzzy Relations, Linguistic Variables. **Membership Functions**: Introduction, Features & Fuzzification, Methods of Membership Value Assignment; Defuzzification. **Fuzzy Systems:** Crisp Logic, Predicate Logic, Fuzzy Logic; Fuzzy Rule Base and Approximate Reasoning, Fuzzy Quantifiers; Fuzzy Inference Systems, Fuzzy Decision Making, Fuzzy Logic Control System; Fuzzy Expert Systems.	15
III	**Neural Networks**: Fundamental Concepts, Basic Models and Architecture; Machine Learning Using Neural Networks; Associative Memory Networks and their Applications. **Supervised Learning Neural Networks**: Perceptron Networks, Radial Basis Function Networks: Back Propagation Neural Network: Architecture, Learning, Applications, & Research Directions; The Boltzman Machine. **Unsupervised Learning Networks:** Competitive Learning networks; Kohonen Self-Organizing Net-	15

	works; Hebbian learning; The Hopfield Network; Counter propagation Networks; Adaptive Resonance Theory: Introduction, Architecture, & Applications; Feed forward Networks; Reinforcement Learning.	
IV	**Genetic Algorithms:** Introduction to Genetic Algorithms and their Terminology; Traditional Optimization and Search Techniques vs Genetic Algorithm; Operators in Genetic Algorithms: selection, crossover, mutation; Problem Solving using Genetic Algorithm; Classification of Genetic Algorithms; Holland’s Schemata theorem; Advantages and Limitations of Genetic Algorithm; Applications of Genetic Algorithm.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

1) S. N. Sivanandam& S. N. Deepa, Principles of Soft Computing, Wiley - India.
2) Jyh Shing Roger Jang, Chuen Tsai Sun, EijiMizutani, Neuro Fuzzy and Soft Computing, Prentice Hall
3) S.Rajasekaran and G.A.VijayalakshmiPai, Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications, Prentice-Hall of India Pvt. Ltd.
4) George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall.
5) George J. Klir, Ute St. Clair, Bo Yuan, Fuzzy Set Theory: Foundations and Applications Prentice Hall.
6) Simon O. Haykin, Neural networks: a comprehensive foundation, Pearson Education.
7) Mitchell Melanie, An Introduction to Genetic Algorithm, Prentice Hall
8) Goldberg D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Pearson Education.
9) Ahmad Lotfi, Jonathan Garibaldi, Applications and Science in Soft Computing, Springer.
10) Rajkumar Roy, Mario Koppen, Soft Computing and Industry: Recent Applications, Springer.
11) James A. Freeman, David M. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques, Pearson Education India.
12) Du, Ke-Lin, Swamy, M. N. S., Search and Optimization by Metaheuristics: Techniques and Algorithms, Springer
13) OmidBozorg-Haddad, Mohammad Solgi, Hugo A. Loaiciga, Meta-heuristic and Evolutionary Algorithms for Engineering Optimization, Wiley

With effect from the Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	4th
Name of the Course	Mobile Computing
Course Code	M24- CAP -409
Course Type	DEC-5
Level of the course (As per Annexure-I	500-599
Pre-requisite for the course (if any)	-
Course Objectives	The objective of this course is to provide students with a solid foundation in the concepts, architecture, and technologies that drive mobile computing systems. The course explores the evolution of mobile communication, wireless networks, and mobile devices, highlighting the design and implementation of mobile applications in a distributed and heterogeneous environment. Students will gain insights into mobile operating systems, wireless transmission principles, cellular technologies, and security challenges associated with mobile environments. The course also addresses current trends in mobile platforms and development frameworks, enabling students to analyze, design, and build efficient and secure mobile computing solutions suited to modern business and societal needs.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1:** Demonstrate an understanding of the fundamental concepts, architectures, and communication models of mobile computing, including various mobile and wireless communication technologies such as GSM, GPRS, 3G/4G, LTE, and mobile IP. **CLO 2**: Describe the Android platform architecture, development tools (SDK, AVD, DDMS), and the Android application lifecycle, and set up a basic Android development environment using Android Studio. **CLO 3**: Design and implement user interfaces for Android applications using various layout managers and UI components, and utilize intents for inter-component communication and interaction within applications. **CLO 4**: Develop Android applications that utilize services, fragments, location-based features, persistent storage mechanisms (internal, external, SQLite), and integrate third-party services such as Google Maps.

	Theory	Practical	Total
Credits	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Mobile Computing: Concepts and scope, Features and applications of mobile computing, Mobile and wireless devices: smartphones, tablets, PDAs, Limitations and challenges of mobile environments, Wireless communication basics: signal propagation, multiplexing, modulation, Overview of mobile communication systems: GSM, GPRS, 3G/4G, LTE, Mobile computing architecture: Internet-based and satellite-based models, Wireless LAN architecture and mobile IP.	15
II	Introduction: Mobile Applications, Characteristics and Benefits, Application Models, Basics of Android, Importance and scope, Android Versions, Features of Android, Android Architecture, Android Stack, Android Applications Structure, Android Emulator, Android SDK, Overview of Android Studio, Android and File Structure, Android Virtual Device Manager, DDMS, LogCat, Understanding Activities (Activity Life Cycle).	15
III	Android User Interface: Measurements –Device and pixel density independent measuring units. Layouts –Linear, Relative, Grid and Table Layouts. User Interface (UI) Components –Editable and non-editable Text Views, Buttons, Radio and Toggle	15

	Buttons, Checkboxes, Spinners, Dialog and pickers, List View, Spinner View. Intents and Broadcasts: Intent –Using intents to launch Activities, Types of Intent, Passing data to Intents, using Intent to dial a number or to send SMS.	
IV	Services- Callbacks and Override in application, Application Signing, API keys for Google Maps, Publishing application to the Android Market. Fragments –Creating fragments, Lifecycle of fragments, adding, removing and replacing fragments with fragment transactions Location and Mapping: location based services, Google Maps activity Persisting Data to files: Saving to Internal Storage, Saving to External Storage Introduction to SQLite database: creating and opening a database, creating tables, content provider	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books

1) Raj Kamal, Mobile Computing, Oxford University Press.
2) Asoke K. Talukder, Hasan Ahmed, and Roopa R. Yavagal, Mobile Computing: Technology, Applications and Service Creation, McGraw-Hill Education.
3) Jochen Schiller, Mobile Communications, Pearson Education.
4) Reto Meier and Ian Lake, Professional Android, Wrox.
5) Wei-Meng Lee, Beginning Android Programming with Android Studio, Wiley.

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4^{th}		
Name of the Course	Cloud Computing and IoT		
Course Code	M24- CAP -410		
Course Type	DEC-5		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course introduces the fundamentals of cloud computing and IoT, covering architectures, service models, virtualization, and enabling technologies. Students will explore cloud services, security, and compliance, as well as IoT components, communication protocols, and ecosystem design. Hands-on experience with platforms like Arduino and Raspberry Pi prepares students to build and manage cloud-integrated IoT systems for real-world applications.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO-1** Understand core issues of cloud computing and enabling technologies. **CLO-2** Learn services provided on cloud computing platforms & security challenges. **CLO-3** Understand concepts, architecture, applications & networking technologies of IoT. **CLO-4** Learn IoT-oriented communication protocols and security concerns.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	**Cloud Computing:** Definition and Characteristics of Cloud Computing, Evolution and Enabling Technologies, Benefits & Challenges, NIST Reference Architecture of Cloud Computing; Deployment Models, Service Models, Service Oriented Architecture (SOA). **Virtualization:** Virtualization and Its Role in Cloud Computing; Benefits & Drawbacks of Virtualization, Types of Virtualization, Server Virtualization, Hypervisor-Based Approaches, Virtualization Of - Operating System, Platform, CPU, Network, Application, Memory and I/O Devices.	15
II	**Cloud Computing Services & Applications:** Cloud Computing Platforms; Compute Services, Storage Services, Database Services, Applications Services, Queuing Services, E-Mail Services, Notification Services, Media Services, Content Delivery Services, Analytics Services, Deployment& Management Services, Identity & Access Management Services and their Case Studies. **Cloud Security and Compliance:** Security Challenges in The Cloud, Data Protection and Privacy, Compliance and Regulatory Issues.	15
III	**Internet of Thing (IoT):** Definition and Characteristics of IoT, Key Components of IoT (Sensors, Actuators, Devices). IoT Ecosystems and Architecture, Conceptual Framework, Common Applications of IoT. Modified OSI Model for IoT/M2M Systems, M2M Vs IoT. **Iot Networking Technologies:** NFC, RFID, Bluetooth BR/EDR and Bluetooth Low Energy, Zigbee, WiFi.	15
IV	**Iot Communication Technologies & Security Issues:** Constrained Nodes, Constrained Networks, Types of Constrained Devices, Low Power and Lossy Networks. Security Issues and Challenges in IoT. **IoT Protocols** – 6LoWPAN, QUIC Protocol; Data Protocols - MQTT, MQTT-SN, CoAP, AMQP. Introduction to Arduino and Raspberry Pi Boards.	15
	Total Contact Hours	60

Suggested Evaluation Methods

Internal Assessment: 30	End Term Examination: 70

➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

Textbook

1) Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing – Principles and Paradigms, Wiley India Pvt. Ltd.
2) Sudip Misra, Anandarup Mukherjee, Arijit Roy, Introduction to IOT, Cambridge
3) Arshdeep Bahga, Vijay Madisetti, Cloud Computing – A Hands-on Approach, University Press.
4) Raj Kamal, Internet of Things - Architecture and Design Principles, McGraw Hills

Reference Books

1) Mayur Ramgir, Internet of Things – Architecture, Implementation and Security, Pearson
2) Kai Hwang, Geoffrey C.Fox, and Jack J. Dongarra, Distributed and Cloud Computing, Elsevier India Private Limited
3) Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Cloud computing: A practical Approach, McGraw Hill
4) Vijay Madisetti and Arshdeep Bahga, Internet of Things (A Hands-on Approach), VPT

DEC-5 Principles of Programming Languages

With effect from the Session: 2024-25			
Part A - Introduction			
Name of the Programme	MCA		
Semester	4^{th}		
Name of the Course	Principles of Programming Languages		
Course Code	M24-CAP-411		
Course Type	DEC-5		
Level of the course (As per Annexure-I	500-599		
Pre-requisite for the course (if any)	-		
Course Objectives	This course offers a foundational understanding of programming languages, covering their evolution, paradigms, and core components. Topics include formal language theory, parsing, language semantics, type systems, and program verification. Students will explore object-oriented concepts, control structures, memory management, and advanced features like parallel programming and scripting languages, gaining both theoretical and practical insights.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO I:** Students will be able to analyze the structure and translation of programming languages using formal methods such as BNF grammars and recursive descent parsing, and understand the role of translators and binding times in language implementation. **CLO II:** Students will understand the theoretical foundations of programming languages through the Chomsky hierarchy, and be able to design and interpret finite state automata and context-free grammars while applying semantic models like attribute grammars and denotational semantics. **CLO III:** Students will be able to implement and compare object-oriented concepts such as abstract data types, inheritance, and polymorphism, and evaluate various sequence control mechanisms used in modern programming languages. **CLO IV:** Students will understand and apply different parameter passing techniques, memory management strategies, and scoping rules, and explore advanced programming topics such as parallelism, exception handling, scripting languages, and distributed processing.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		
Part B- Contents of the Course			

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Preliminaries: History, Impact of Programming Paradigms, Role of Programming Languages, Good Language, Effects of Programming Environment, Translators and virtual architectures, Binding and Binding time, Language Syntax, Analysis of Program, Synthesis of Object program, Formal translation models: BNF Grammars, General parsing, Language translation, Recursive descent parsing.	15
II	Formal languages and automata: The Chomsky hierarchy of formal languages, regular grammars, Regular expressions, Finite State Automata, Context-free grammars, Pushdown automata, Ambiguous grammars. Language Semantics: Attribute grammars, Denotational semantics, Program verification and validation, Data objects, variables, constants, data types, declaration, type checking, type casting, type promotion, Enumerators, Composite data types.	15
III	Object Orientated concepts: Structured data types, Abstract data types, Information hiding, Subprogram concepts, Good program design, Type definitions, Type equivalence, Inheritance, Derived classes, Abstract classes, Polymorphism, Inheritance and software reuse. Sequence control: Implicit and explicit sequence control, Sequence control within arithmetic expressions, sequence control between statements, sequencing with non-arithmetic expressions, Subprogram Sequence	15

	control.	
IV	Miscellaneous topics: Parameter passing techniques, Static & Dynamic Scoping, Storage of variables, Static storage, Heap Storage management, Distributed Processing, Exceptions and Exception handlers, Co-routines, Scheduled subprograms, Parallel programming, Processor design, Hardware and Software architectures, Network Programming, Evolution of scripting languages, Applets, XML.	15
	Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
➢ **Theory**	**30**	➢ **Theory**	**70**
➢ Class Participation:	5	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	10		
➢ Mid-Term Exam:	15		

Part C-Learning Resources

1) Pratt T.W., Zelkowitz M.V., Gopal T.V., Programming Languages Design and Implementation, Pearson Education.
2) Sebesta W. Robert, Concepts of Programming Languages, Pearson Education.
3) Appleby Doris &VandeKopple J. Julius, Programming Languages-Paradigm and practice, Tata McGraw Hill.
4) Sethi Ravi, Programming Languages: Concepts & Constructs, Pearson Education
5) Scott M., Programming Language Pragmatics, Elsevier India.

With effect from the Session: 2024-25

Part A - Introduction

Name of Programme	MCA		
Semester	4th		
Name of the Course	Cyber Security Fundamentals		
Course Code	M24-CAP-413		
Course Type	EEC		
Level of the course	500-599		
Pre-requisite for the course (if any)	None		
Course Objectives	The objective of this course is to introduce students to the fundamental concepts, principles, and practices of cyber security. It aims to develop a foundational understanding of cyber threats, vulnerabilities, attack types, and security mechanisms. The course covers a broad spectrum of topics including system and network security, cryptographic techniques, risk management, cyber laws, and ethical practices. Students will learn about current trends in cyber attacks and the defensive techniques used to secure data, applications, and infrastructure in both individual and organizational contexts. The course emphasizes a practical understanding of security tools and techniques, fostering the skills necessary to protect information systems and ensure confidentiality, integrity, and availability of digital assets.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO 1**: Students will be able to explain the key concepts of cyber security, identify common types of cyber threats, and understand the foundational principles of securing digital systems. **CLO 2**: Students will demonstrate knowledge of cryptographic techniques and security mechanisms used to ensure data confidentiality, integrity, and secure communication. **CLO 3**: Students will analyze system and network security vulnerabilities and apply basic security measures to protect digital infrastructure. **CLO 4**: Students will understand the legal and ethical aspects of cyber security and recognize current trends and professional opportunities in the field.		
Credits	Theory	Tutorial	Total
	2		2
Teaching Hours per week	2		2
Internal Assessment Marks	15		15
End Term Exam Marks	35		35
Max. Marks	50		50
Examination Time	3		3

Part B- Contents of the Course

<u>Instructions for Paper- Setter:</u> The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	Topics	Contact Hours
I	Introduction to Cyber Security: Importance of cyber security in the digital era, objectives and scope, fundamental concepts such as assets, threats, vulnerabilities, risk, attack vectors, and security controls. Types of cyber attacks: malware, phishing, social engineering, denial of service, ransomware, man-in-the-middle attacks. Principles of cyber defense and the CIA triad (confidentiality, integrity, availability).	8
II	Security Mechanisms and Cryptography: Authentication methods, password policies, access control models, firewalls, intrusion detection and prevention systems (IDPS), virtual private networks (VPNs), antivirus tools. Basics of cryptography: symmetric and asymmetric encryption, digital signatures, hashing algorithms, key management, public key infrastructure (PKI), and applications of cryptography in securing data.	7

III	Network and System Security: Security issues in operating systems and applications, endpoint security, patch management, secure configuration, wireless network security, securing web applications, browser security, and email security. Network security architecture, IP security, and secure communication protocols such as SSL/TLS and HTTPS.	8
IV	Cyber Laws, Ethics, and Emerging Trends: Overview of cyber laws in India and global regulations (IT Act, GDPR), ethical hacking, digital forensics, data privacy and governance. Introduction to security frameworks and standards (ISO 27001, NIST, OWASP). Emerging cyber threats in IoT, cloud computing, AI, and social media.	7
Total Contact Hour		**30**

Suggested Evaluation Methods			
Internal Assessment: 15		**End Term Examination: 35**	
➢ **Theory**	**15**	➢ **Theory:**	**35**
➢ Class Participation:	4	Written Examination	
➢ Seminar/presentation/assignment/quiz/class test etc.:	4		
➢ Mid-Term Exam:	7		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) William Stallings, Network Security Essentials: Applications and Standards, Pearson Education.
2) Nina Godbole and Sunit Belpure, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley India.
3) Chuck Easttom, Computer Security Fundamentals, Pearson Education.
4) Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill Education.
5) Rao B. Ramesh and Govindarajulu, Cyber Security Fundamentals, Wiley.

Dissertation / Project

With effect from the Session: 2024-25	
Name of Programme	MCA
Semester	4th
Name of the Course	Dissertation / Project
Course Code	M24-CAP-414
Course Type	Dissertation / Project
Level of the course	500-599
Pre-requisite for the course (if any)	
Course Objectives	The objective of the Dissertation/Project course is to provide the students with an opportunity to undertake an in-depth investigation of a contemporary research problem or to apply their knowledge and skills in a real-world industrial setting. The course is designed to develop independent research aptitude, analytical and problem-solving skills, and the ability to design and implement effective computational solutions. Students may choose to work on a research-oriented topic leading to a dissertation report, or on an industry-based project culminating in a project report. The emphasis is on fostering innovation, critical thinking, technical competence, and professional reporting and presentation skills, thereby preparing students for higher research, entrepreneurial ventures, or responsible professional roles in the IT industry.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	**CLO1:** Demonstrate the ability to identify, formulate, and analyze a research problem or industrial requirement in the domain of computer applications. **CLO2:** Apply advanced concepts, tools, and methodologies to design and implement effective computational solutions. **CLO3:** Develop independent research, project management, and technical documentation skills with professional standards. **CLO4:** Present findings and outcomes effectively through a well-structured dissertation/project report and oral defense before experts.
Credits	12
Max. Marks	300

Guidelines for Examiners for Dissertation:

1. Relevance of Topic – Assess whether the chosen problem/project is significant, original, and relevant to the field of Computer Applications.
2. Literature Review & Problem Formulation – Evaluate the depth of background study, clarity in objectives, and problem definition.
3. Methodology & Technical Content – Examine the appropriateness of research methods, tools, algorithms, and technologies employed.
4. Implementation & Results – Assess the quality of system design, implementation, experiments conducted, and accuracy/reliability of results.
5. Report Quality – Assess organization, clarity, formatting, referencing, and overall technical writing of the dissertation/project report.
6. Originality & Innovation – Look for independent thinking, novelty of the solution, and avoidance of plagiarism.
7. Viva-Voce / Presentation – Evaluate student's understanding, ability to defend work, and clarity in oral communication.

Guidelines for Examiners (Industrial Project Evaluation)

1. Problem Identification & Industry Relevance: Assess whether the project addresses a real-world industrial problem and its relevance to current practices/technologies.
2. Project Objectives & Scope: Evaluate clarity of objectives, scope definition, and alignment with industry requirements.
3. System Design & Methodology: Examine the design models, architecture, frameworks, and tools used for problem-solving.
4. Assess quality of coding, integration, testing, and overall execution of the project.
5. Assess clarity, completeness, and professionalism of the project report, including industry-specific standards followed.
6. Viva-Voce / Presentation: Evaluate student's ability to present the project, explain technical decisions, and respond to queries confidently.