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

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

Eligibility Criteria

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

  OR

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

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

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

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

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

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

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

PROGRAM SPECIFIC OUTCOMES (PSOs)

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

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

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

Programme Learning Outcomes (PLOs)

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

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

Two Year MCA Syllabus Structure (CW+CW)/(CW+R): First Three Semesters Are Common

Credit Level	Semester	Course Type	Course Code with Name	Course Level	Credits	Total Credits	Max. Marks			Credit	Contact Hours
							Continuous Assessment	End Semester	Total	L: T: P	
6.0 (CW + CW)	Sem - I	Core	MMCACJP125: Java Programming	400	4	22	28	72	100	4:0:0	60
		Core	MMCACML125: Machine Learning	400	4		28	72	100	4:0:0	60
		Core	MMCACRM125: Research Methodology	400	2		14	36	50	2:0:0	30
		DCE-I	MMCADAD125: Advanced Data Structures MMCADCG125: Computer Graphics MMCADMI125: Management Information System MMCADSE125: Software Engineering	400	4		28	72	100	4:0:0	60
		DCE-II	MMCADDS125: Advanced Database Systems MMCADAI125: Artificial Intelligence MMCADBC125: Block Chain Technologies MMCADCS125: Cyber Security & Digital Forensics	400	4		28	72	100	4:0:0	60
		Lab	MMCALJP125: Java Programming Lab	400	2		14	36	50	0:0:2	60
		Lab	MMCALML125: Machine Learning Lab	400	2		14	36	50	0:0:2	60
	Sem - II	Core	MMCACDA225: Design and Analysis of Algorithms	400	4	22	28	72	100	4:0:0	60
		Core	MMCACMA225: Mobile Application Development	400	4		28	72	100	4:0:0	60
		Core	MMCACRP225: Research and Publication Ethics	400	4		28	72	100	4:0:0	60
		DCE-III	MMCADAO225: Advanced Operating Systems MMCADDI225: Digital Image Processing MMCADDS225: Decision Support Systems MMCADCN225: Cryptography & Network Security	400	4		28	72	100	4:0:0	60
		DCE-IV	MMCADAC225: Advanced Computer Networks MMCADCC225: Cloud Computing MMCADLP225: Linux Programming MMCADTC225: Theory of Computation	400	4		28	72	100	4:0:0	60
		Lab	MMCALMA225: Mobile Application Development Lab	400	2		14	36	50	0:0:2	60
Total (First Year)					44	44	308	792	1100	38:0:6	750 Hrs
6.5 (CW + CW) / (CW + R)	Sem - III	Core	MMCACDS325: Data Science with Python	500	4	22	28	72	100	4:0:0	60
		Core	MMCACWP325: Web Programming	500	4		28	72	100	4:0:0	60
		Core	MMCACSP325: Software Project Management	500	2		14	36	50	2:0:0	30
		DCE-V	MMCADQC325: Quantum Computing MMCADEH325: Ethical Hacking MMCADCV325: Computer Vision MMCADER325: Enterprise Resource Planning	500	4		28	72	100	4:0:0	60
		DCE-VI	MMCADNL325: Natural Language Processing MMCADSQ325: Software Quality Assurance MMCADDL325: Deep Learning MMCADIT325: Internet of Things	500	4		28	72	100	4:0:0	60
		Lab	MMCALDS325: Data Science with Python Lab	500	2		14	36	50	0:0:2	60
		Lab	MMCALWP325: Web Programming Lab	500	2		14	36	50	0:0:2	60
			OPTION – 1 for Semester – IV								
	Sem – IV CW+CW	Core	MMCACRA425: Requirements Analysis & Problem Solving	500	4	20	28	72	100	0:4:0	60
		Core	MMCACSD425: Software Design Patterns & Architecture	500	4		28	72	100	0:4:0	60
		Core	MMCACSI425: Software Design & Implementation	500	4		28	72	100	0:0:4	120
		Core	MMCACST425: Software Test Case Design and Implementation	500	4		28	72	100	0:0:4	120
		Core	MMCACDI425: Dissertation	500	4		28	72	100	0:4:0	60
			OPTION – 2 for Semester - IV								
	Sem – IV CW+R	Internship	MMCAIID425: Internship Dissertation	500	20	20	140	360	500	0:12:8	420
Total (Second Year)					42	42	294	756	1050	18:16:12	810 Hrs
TOTAL CREDITS (AGGREGATE OF 4-SEMESTERS)					*86*	*86*	*602*	*1548*	*2150*	*56:16:18*	*1560 Hrs*

SEMESTER - I

COURSE TITLE:JAVA PROGRAMMING

Course Code: MMCACJP125						**Examination Scheme**	
Total number of Lecture Hours: 60						**External**	**72**
						Internal	**28**
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	

COURSE LEARNING OUTCOMES (CLO):

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

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

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

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

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

UNIT 3: Advanced Java Concepts	15- Hrs
Packages: Creating and Importing Packages. CLASSPATH variable. static import. **Strings:** Mutable and Immutable Strings. Creating Strings. Operations on Strings. **Threads:** Creating Threads in Java. Java Thread Lifecycle. Multithreading in Java: Synchronization and Inter- process communication (IPC) in Threads. **I/O Streams:** Byte, Character, Buffered, Data, and Object Streams. Standard Streams. File I/O Basics, Reading and Writing to Files. Serializing Objects.	
UNIT 4: Java GUI Programming and Networking	**15-Hrs**
Event-Driven Programming: Java 1.1 Event Delegation Model – Source object, Event object and Listener object. Methods associated with Source, Event and Listener objects. Low-level vs Semantic events. Adapter classes, Inner classes, and Anonymous Inner classes. Adding GUI elements to Applet. **Networking Classes and Interfaces:** TCP/IP Server Sockets in Java. Developing simple networking applications in Java like File transfer, Chatting, etc.	

CLO-PLO Matrix for the Course

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

Textbooks

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

Reference Books

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

COURSE TITLE: Machine Learning

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

COURSE LEARNING OUTCOMES (CO):

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

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

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

CLO-PLO Matrix for the Course

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

Textbooks

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

Reference Books

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

DCE-I

COURSE TITLE: Advanced Data Structures									
Course Code: MMCADAD125						**Examination Scheme**	**T**	**P**	
Total number of Lecture Hours: 60						**External**	**72**	-	
						Internal	**28**	-	
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**		

COURSE LEARNING OUTCOMES (CLO):

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

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

Unit IV: Advanced Data Structures and Algorithms	15 Hrs.
Minimum spanning trees, Shortest Path Algorithms in Graphs, Eulerian Tour, Hamiltonian Tour Hashing: Direct Address Tables, Hash Table, Different Hash functions, resolving collisions, rehashing, Heap Structures, Binomial Heaps, Leftist Heaps. File Organizations: Sequential File Organization, Relative File Organization, Indexed Sequential File Organization, Multiple Key File Organizations: Inverted File and Multi-List Organizations	

CLO-PLO Matrix for the Course

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

Textbooks

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

Reference Books

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

COURSE TITLE: Computer Graphics									
Course Code: MMCADCG125						**Examination Scheme**	**T**	**P**	
Total number of Lecture Hours: 60						**External**	**72**	-	
						Internal	**28**	-	
Lecture (L):	**4**	**Practical(P):**	**0**	**Tutorial (T):**	**0**	**Total Credits**	**4**		

COURSE LEARNING OUTCOMES (CLO):

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

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

LEVEL OF CO-PO MAPPING TABLE

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

Textbooks

1. Hearn and Baker, "Computer Graphics with OpenGL": 4th Edition (2022), Donald Hearn, M. Pauline Baker, Warren Carithers, Pearson
2. Ze-Nian Li and Mark S. Drew, "Fundamentals of Multimedia": 3rd Edition (2021), Springer.
3. **W.M. Newman and Sproull, "Principles of Interactive Computer Graphics"** McGraw-Hill Education; 3rd Edition, 2023.

Reference Books

1 **Steven Harrington, "Computer Graphics: A Programming Approach"** McGraw-Hill Education; 2nd Edition, 2021.

2 **Plastock and Kelley, "Schaum's Outline of Theory and Problems of Computer Graphics"** McGraw-Hill Education; 2nd Edition, 2022.

3 **David F. Rogers and J. Alan Adams, "Procedural Elements of Computer Graphics"** McGraw-Hill Education; 3rd Edition, 2021.

4 **David F. Rogers and J. Alan Adams, "Mathematical Elements of Computer Graphics"** McGraw-Hill Education; 3rd Edition, 2022.

5 **James D. Foley, Andries van Dam, et al., "Computer Graphics: Principles and Practice"** Pearson; 4th Edition, 2023.

6 **Sinha and Udai, "Computer Graphics"** Tata McGraw-Hill Education; 2nd Edition, 2022

COURSE TITLE: Management Information System

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

COURSE LEARNING OUTCOMES (CLO):

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

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

LEVEL OF CO-PO MAPPING TABLE

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

Textbooks

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

Reference Books: -

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

COURSE TITLE: Software Engineering									
Course Code: MMCADSE125							Examination Scheme	T	P
Total number of Lecture Hours: 60							External	72	-
							Internal	28	-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits			4

COURSE LEARNING OUTCOMES (CLO):

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

Course Content	TEACHING HOURS
UNIT 1: Fundamentals of Software Engineering	**15 Hrs**
Concept and Nature of Software: Concept and Nature of Software, Software Crisis, Software Engineering – Concept, Goals and Challenges, Software Engineering Approach. Software Development Process, Process Models - Waterfall Model, Evolutionary and Throwaway Prototyping Model, Incremental and Iterative Models, Spiral Model, Agile Process Model, Component based and Aspect Oriented development Software Process and Project Measurement: Measures, Metrics and Indicators, Size - Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model.	
UNIT 2: Requirements Engineering	**15 Hrs**
Introduction to Requirements Engineering - Why, What and Where. Requirements Types: functional and nonfunctional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques. Requirement Analysis and Modelling, Requirements prioritization, verification, and validation.	
UNIT 3: Design Engineering	**15 Hrs**
Basics of Design Engineering - Abstraction, Architecture, Patterns, Separation of concerns, Modularity, Functional Independence, refinement, Refactoring. Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology. Object-Oriented Design - Design Concepts, Design Methodology, Object-oriented analysis and design modeling using Unified Modeling Language (UML), Dynamic & Functional Modeling, Design Verification.	
UNIT 4: Software Testing and Reliability	**15 Hrs**

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

CLO-PLO Matrix for the Course

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

Textbooks

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

Reference Books

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

DCE-II

COURSE TITLE: Advanced Database Systems										
Course Code: MMCADDS125						Examination Scheme	T		P
Total number of Lecture Hours: 60						External	72		-
						Internal	28		-
Lecture (L):	4	Practical (P):	-	Tutorial (T):	-	Total Credits			4

COURSE LEARNING OUTCOMES (CLO):

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

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

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

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

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

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

CLO-PLO Matrix for the Course

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

Textbooks:

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

Reference Books:

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

COURSE TITLE: Artificial Intelligence							
Course Code: MMCADAI125					**Examination Scheme**	**T**	**P**
Total number of Lecture Hours: 60					**External**	**72**	**-**
					Internal	**28**	**-**
Lecture(L):	**4**	**Practical (P):**	**-**	**Tutorial(T):**	**Total Credits**	**4**	

COURSE LEARNING OUTCOMES(CO):

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

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

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

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

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

CLO-PLO Matrix for the Course

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

Textbooks
1. "**Artificial Intelligence: A Guide to Intelligent Systems**" by **Michael Negnevitsky, Latest Edition, 2020.** 2. *"Artificial Intelligence: A Modern Approach"* by Stuart Russell and Peter Norvig, **4th Edition,** 2020. 3. "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell, **Latest Edition,** 2019
Reference Books
1. "**Artificial Intelligence**" by **Elaine Rich, Kevin Knight, and Shivashankar B. Nair, 4th Edition, 2021**. 2. "Artificial Intelligence: Foundations of Computational Agents" by Michael Wooldridge, Ist Edition, 2021. 3. **"Nature-Inspired Optimization Algorithms"** by **Saeid Aziznejad, Gholamreza Z. Naderpour, and Mohammad A. H. Sadeghi, Ist Edition, 2019.**

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

COURSE LEARNING OUTCOMES (CLO):

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

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

Tests .	
Unit 4: Advanced Topics and Future Directions in Blockchain	**15 Hrs**
Blockchain in IoT - Use Cases, Challenges, and Solutions. Blockchain and Big Data - Integration, Analytics, and Use Cases. Blockchain in AI - Synergies, Applications, and Challenges. Blockchain and Cloud Computing - Decentralized Cloud Solutions. Green and Sustainable Blockchain Technologies. Quantum Computing and its Impact on Blockchain . Future Directions - Web 3.0, Decentralized Identity, and Tokenization of Assets.	

CLO-PLO Matrix for the course

Unit wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Average (CLO)
MMCADBC125.1	3	2	2	2	3	2	2	2	2	3	**2.3**
MMCADBC125.2	3	3	2	2	3	2	2	2	3	3	**2.5**
MMCADBC125.3	3	3	3	2	3	1	2	3	2	3	**2.5**
MMCADBC125.4	3	2	2	2	3	2	3	3	3	2	**2.5**
Average (PLO)	**3.0**	**2.5**	**2.25**	**2.0**	**3.0**	**1.75**	**2.25**	**2.5**	**2.5**	**2.75**	**2.45**

Textbooks:

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

Reference Books:

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

COURSE TITLE: Cyber Security and Digital Forensics

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

COURSE LEARNING OUTCOMES (CLO):

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

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

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

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

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

CLO-PLO Matrix for the Course

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

Textbooks:
1. E. Casey, Handbook of Digital Forensics and Investigation, Academic Press, 2010.
Reference Books:
1. B. Carrier, File System Forensic Analysis, Addison-Wesley, 2005. 2. J.R. Vacca and K. Rudolph, System Forensics, Investigation and Response, Jones and Bartlett Learning, 2011. 3. M. T. Britz, Computer Forensics and Cyber Crime, Pearson, 2013.

COURSE TITLE: Research Methodology

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

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

COURSE LEARNING OUTCOMES(CLO):

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

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

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

CLO-PLO Matrix for the Course

Unit-Wise CLOs	PLOs										
	1	2	3	4	5	6	7	8	9	10	Avg (CLO)
MMCACRM125.1	3	2	2	2	3	3	3	2	3	2	2.5
MMCACRM125.2	3	3	3	3	3	2	3	2	3	2	2.7
Avg (PLO)	3.0	2.5	2.5	2.5	3.0	2.5	3	2	3	2	2.6

Textbooks

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

ReferenceBooks

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

COURSE TITLE: JAVA PROGRAMMING (Lab)

Course Code: MMCALJP125						**Examination Scheme**	
Total number of Lab Hours: 60 hours						**External**	**36**
						Internal	**14**
Lecture (L):	**-**	**Practicals (P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**2**

COURSE LEARNING OUTCOMES (CLO):

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

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

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

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

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

- Write a Java program that creates a Class namely A that has a private instance variable and method, a protected instance variable and method, a default instance variable and method, and a public instance variable and method. Create another Class say B that inherits from A.
  - Show that all except private members are inherited.
  - Show that an inherited instance variable can be shadowed (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
  - Show that an inherited method can be overridden (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.
  - Show that the reference variable of type A or B can't access an overridden method of A in the Object of B.
  - Show that the reference variable of type A can access a shadowed data member of A in the Object of B.

- **Week 5:**
  - Write a Java program that creates a Class in which a method asks the user to input 2 integer values, and calls another member function (say div()) to divide the first inputted number by the second number (by passing them as parameters). Handle an exception that can be raised in div() when the denominator equals zero (use try-catch statement).
  - Modify the above Java program so that it also creates a Custom Exception that is thrown by div() when the denominator value is 1 (use throw). Handle the exception. c.
  - Modify the above Java program so that the exception-handling in not performed by div() rather it only species all the possible exceptions it may throw (use throws). And, the method that calls div() does the exception handling.

- **Week 6:**
  - Create a Java Package (say pack1) that contains 3 Classes (say A, B and C). Write a Java program that uses this package after setting the CLASSPATH variable. Following scenarios must be considered individually:
    - Importing the whole package (all the 3 classes)
    - Importing only specific class (say Class A only)
  - Create another Package (say pack2) that contains same number of classes, and same definition for each class, as that of pack1. Write a Java program that imports all classes from both pack1 and pack2 while ensuring that the name conflicts are not encountered while accessing any of these classes.

- **Week 7:**
  - Write a Java program to count the number of words in a string that is passed as a command line argument.
  - Write a Java program to check whether a string is palindrome or not.
  - Write a Java program to count the total number of occurrences of a given character in a string.
  - Write a Java program to convert a string to char array.

- **Week 8:**
  - Write a Java program that creates a Class that extends a Thread class. Create 3 objects of the class, each starting a new thread and each thread displaying "I am Thread: " in an infinite loop. The displayed text must be suffixed by the unique name of the thread.
  - Write a Java program that creates a Class that implements interface Runnable, and does the same as the above program.

- Write a Java program to implement a solution for producer-consumer problem using synchronization and inter-process communication in Threads.

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

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

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

COURSE TITLE: Machine Learning Lab

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

COURSE LEARNING OUTCOMES (CO):

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

Practical's

Week 1: Python and ML Tools Setup

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

Week 2: Google Colab

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

Week 3: Data Handling and Visualization

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

Week 4: Data Preprocessing

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

Week 5: Data Analysis

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

Week 6: Classification with Decision Tree and KNN

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

Week 7: Exploring KNN Parameters

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

Week 8: Regression Techniques

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

Week 9: Model Evaluation

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

Week 10: Clustering with K-Means

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

Week 11: Advanced Classification Tasks

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

Week 12: Dimensionality Reduction using PCA

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

Week 13: Naïve Bayes Algorithm

Implement Naïve Bayes from scratch and answer the following:

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

Week 14: Model Evaluation and Real-World Applications

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

CLO-PLO Matrix for the Course

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

Textbooks

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

Reference Books

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