Master of Computer APPLICATION (MCA)

Session2024-25

divya

Program Outcomes for MCA

PO 1: Apply mathematics and computing fundamental and domain concepts to find out the solution of defined problems and requirements. (Computational Knowledge)

PO 2: Use fundamental principle of Mathematics and Computing to identify, formulate research literature for solving complex problems, reaching appropriate solutions. (Problem Analysis)

PO 3: Understand to design, analyze and develop solutions and evaluate system components or processes to meet specific need for local, regional and global public health, societal, cultural, and environmental systems. (Design/Development of Solutions)

PO 4: Use expertise research-based knowledge and methods including skills for analysis and development of information to reach valid conclusions. (Conduct Investigations of Complex Computing Problems)

PO 5: Use expertise research-based knowledge and methods including skills for analysis and development of information to reach valid conclusions. (Conduct Investigations of Complex Computing Problems)

PO 6: Exhibiting ethics for regulations, responsibilities and norms in professional computing practices. (Professional Ethics)

PO 7: Enlighten knowledge to enhance understanding and building research, strategies in independent learning for continual development as computer applications professional. (Lifelong Learning)

PO 8: Establishing strategies in developing and implementing ideas in multi- disciplinary environments using computing and management skills as a member or leader in a team. (Project Management and Finance)

PO 9: Contribute to progressive community and society in comprehending computing activities by writing effective reports, designing documentation, making effective presentation, and understand instructions. (Communication Efficacy)

PO 10: Gain confidence for self and continuous learning to improve knowledge and competence as a member or leader of a team. (Individual and Teamwork)

PO 11: Gain confidence for self and continuous learning to improve knowledge and competence as a member or leader of a team. (Individual and Teamwork)

Program Specific Outcomes (PSO):

1) To guide and channelize the transformation process of every management graduate by providing in-depth knowledge of business management and entrepreneurship embedded wit h ethics and a sense of social commitment and to make them to strive towards personal victory and value creation to society.

2) To ignite a passion for multidisciplinary approach for problem solving, critical analysis and decision making by giving due importance for lateral thanking so that management graduates see things from a perspective which are not just simple but effective. 3) Student will able to know various issues, latest trends in technology development and thereby innovate new ideas and solutions to existing problems.

Program Education Outcome (PEO)

PSO1: To prepare the graduates as successful professionals ready for Industry, Government sectors, Academia, Research, Entrepreneurial Pursuit and Consultancy firms.

PSO2 :To prepare the graduates with Ethical Attitude, Effective Communication Skills and admit themselves as ethical and responsible cit izens with social commitments.

PSO3: To prepare the graduates with excellent computing ability so that to Comprehend, Analyse, Design and Create computing solutions for the real-time problems.

PSO4: In order to enhance programming skills of the young IT professionals, the concept of project development in using the technologies learnt during the semester has been introduced.

PSO5: To enhance knowledge in robotics, provide experimental hardware equipment for teaching the basics of robotics, robot dynamics and control, and robot system design and application.

PSO6: To enhance logical ability and programming concepts by implementing programming lab.

PSO7: Preparing students for future aspects by building and improving their creativity, social awareness, and general knowledge.

PSO8: Encouraging students to convert their start-up idea to reality by implementing.

PSO9: Ability to understand the changes or future trends in the field of computer application.

PSO10: Ability to identify, formulate, analyse and solve problems of programming using different languages.

PAHER University, Udaipur
Faculty of Computer Science
LMCA (Master of Computer Applications)- 2Yr Course
SEMESTER-I

S. No.	Code	Subject Name	Period			Evaluation Scheme			
			L	T	P	ISE	ESE	Total	Credit
1.	LMCA:101	Block Chain Technology	3	1	0	30	70	100	4
	LMCA:102	Mathematical Foundations In Computer Science	3	1		30	70	100	4
2.	LMCA:103	It Application And Data Security	3	1	0	30	70	100	4
3.	LMCA:104	Programming Fundamentals Using Python	3	1	0	30	70	100	4
4.	LMCA:105	Programming Through C++	3	1	0	30	70	100	4
5.	LMCA:106	Web Programming	3	1	0	30	70	100	4
6.	LMCA:107	Programming Through C++ Lab	0	0	4	30	70	100	2
7.	LMCA:108	Web Programming Lab	0	0	4	30	70	100	2
8.	LMCA:109	Programming Fundamentals Using Python Lab	0	0	4	30	70	100	2
10	LMCA:110	Seminar	0	0	4	30	70	100	2
			18	6	16			1000	32

Course Code	**LMCA: 101**
Course Title	**Block chain Technology**
Credits	4
Course Objective(s)	Students will demonstrate a solid understanding of foundational AI concepts and historical developments. They will be able to identify and apply core AI techniques, such as machine learning and natural language processing, to solve real-world problems. Additionally, students will recognize the ethical implications and societal impacts of AI technologies. They will gain practical experience with AI tools and frameworks, enabling them to complete projects effectively. Furthermore, students will explore interdisciplinary applications of AI across various fields and stay informed about current trends and future directions in AI research and development.
Course Outcome (CO)	**CO1**: **Foundational Knowledge**: Introduce the fundamental concepts and history of artificial intelligence. **CO2**: **Core Techniques**: Explore key AI techniques such as machine learning, natural language processing, and computer vision. **CO3**: **Problem-Solving Skills**: Develop problem-solving and analytical skills through AI methodologies. **CO4**: **Ethics and Impact**: Understand the ethical implications and societal impact of AI technologies. **CO5**: **Hands-on Experience**: Gain practical experience with AI tools and frameworks through projects and exercises.

LMCA-101 Block chain Technology

Unit I

Overview of Block chain Technology: Defining Block chain and Distributed Ledger, Block chain Properties Decentralized, Transparent, and Immutable and secure. Block chain Applications. Types of Block chain: Public, private, and consortium based block chain, when to use, and when not to use Block chain, History of Block chain.

Introduction to computing models and P2P networking: Centralized, Decentralized and Distributed Systems, Decentralization vs. distributed, P2P systems, propertied of P2P systems, P2P communication architecture. P2P network applications: File sharing, P2P network for block chain

Unit II

Foundational Concepts Block chain Data Structure: Cryptographic Hash Functions, Digital Signatures, Public Keys as Identities, Hash Pointers and Hash chain and Merkel tree, Consensus mechanisms **Block chain Characteristics: Decentralized** Identity management, Transactions, incentivising and mining. Distributed Consensus (PoW), Crypto currency as the first block chain application. Mechanics of Bit coin, Bit coin Scripts, Storing and Using Bit coins, Mining in Bit coin.

Unit III

Other Consensus Mechanisms: Proof of storage, proof of stake, proof of deposit, proof of burn, proof of activity. Algorithms for adjusting difficulty and retargeting. Limitations of Bit coin, alternative crypto currencies.

Smart Contracts and Ethereum: History, Purpose and types of smart contracts, Introduction to Ethereum, bit coin vs Ethereum stack. P2P network in Ethereum, consensus in Ethereum, scripts in Ethereum, Smart contracts (Ethereum Virtual Machine). Developing and executing smart contracts in Ethereum. State and data structure in Ethereum.

Unit IV

Private and Consortium based Block chain: Hyper ledger Need for the consortium. Hyper ledger stack, Multi chain block chain. Innovation in Hyper ledger, smart contracts, and distributed applications in hyper ledger.

Unit V

Case studies/ Enabling Technologies and applications: Application of block chain in privacy and security, IoT and smart cities, Business and Industry, Data management, e-Governance

Text Books:

- Imran Bashir, “Mastering Block chain: Distributed Ledger Technology, decentralization, and smart contracts explained”, 2nd Edition, Packt Publishing Ltd, March 2018.
- Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, “Block chain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger”, Packt Publishing Limited, 2018.

Course Code	**LMCA:102**
Course Title	**Mathematical Foundations in Computer Science**
Credits	4
Course Objective(s)	Upon completing the "Mathematical Foundations in Computer Science" course, students will be able to apply key mathematical concepts to solve computational problems, analyze algorithm efficiency, construct rigorous proofs, and utilize mathematical reasoning in areas like data structures and cryptography. They will also enhance their critical thinking skills and be prepared for advanced studies in computer science.
Course Outcome (CO)	**CO1**: **Understanding Mathematical Concepts:** To provide students with a solid understanding of fundamental mathematical concepts that are essential for computer science, including logic, set theory, functions, relations, and combinatory**.** **CO2**: **Problem-Solving Skills**: To develop students' problem-solving abilities by applying mathematical reasoning to analyze and solve computational problems. **CO3**: **Algorithm Analysis:** To introduce students to the mathematical foundations of algorithms, including complexity analysis, recurrence relations, and asymptotic notation. **CO4**: **Hands-on Experience**: Gain practical experience with AI tools and frameworks through projects and exercises. **CO5: Formal Methods:** To familiarize students with formal methods and mathematical proofs, enabling them to construct and validate arguments rigorously**.**

LMCA: 102 Mathematical Foundations in Computer Science

Unit -I

Matrices: Introduction, Rank of Matrix, Solving System of Equations, Inverse of a Matrix, Set theory, Principle of inclusion and exclusion, partitions, Permutation and Combination, Relations, Properties of relations, Matrices of relations, Closure operations on relations, Functions- injective, subjective and objective functions.

Unit- II

Probability: Probability Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes' Theorem and independence problems. Introduction to Statistics- Population, Sample, Variable, Descriptive Statistics-Mean, Mode, Median, Measures of SpreadRange, Inter Quartile Range, Variance, Standard Deviation.

Unit -III

Propositions & Propositional Calculus: Propositions and logical operators, Truth table, Propositions generated by a set, Equivalence and implication, Basic laws, Functionally complete set of connectives, Normal forms, Proofs in Propositional calculus, Predicate calculus.

Unit- IV

Data Representation: Data Representation - Floating point Arithmetic – Addition, Subtraction, Multiplication and Division operation. Pitfall of floating point representation, Errors in numerical computation Iterative Methods, Measurement of Accuracy by using Absolute Error and Relative Error.

Unit-V

Graphs & Trees: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Planar Graphs, Euler's Formula, Spanning Trees

Text Books:

1.Y.N. Singh-Mathematical Foundations of Computer Science(New age International Publishers)

2.Discrete Mathematical Structures with Applications to Computer Science,J. P.Tremblay and P. Manohar,Tata McGraw Hill.

3.Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rdEdition, Tata McGraw Hill. 3. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, K. H. Rosen, 7th Edition, Tata McGraw Hill.

Course Code	**LMCA:103**
Course Title	**IT Application And Data Security**
Credits	4
Course Objective(s)	Upon completing the "IT Application and Data Security" course, students will be able to understand key security principles, identify risks, implement best practices, and use security tools. They will also develop security policies, conduct risk assessments, ensure regulatory compliance, and manage security incidents, preparing them to enhance organizational data security.
Course Outcome (CO)	**CO1**: **Understand Security Principles:** Grasp fundamental concepts of information security, including confidentiality, integrity, and availability. **CO2**: **Identify Threats and Vulnerabilities:** Recognize various types of security threats, vulnerabilities, and risks associated with IT applications and data. **CO3**: **Implement Security Measures:** Learn to apply security measures and best practices to protect IT applications and data from unauthorized access and breaches. **CO4**: **Utilize Security Technologies:** Gain hands-on experience with security technologies such as firewalls, encryption, intrusion detection systems, and access control mechanisms. **CO5: Develop Security Policies:** Develop Security Policies: Understand how to create and implement effective security policies and procedures within an organization

LMCA: 103 It Applications and Data Security

Unit I

Introduction to software development & application security: Basics of programming languages, complied versus interpreted, Programming concepts, Distributed programming, Threats and malware; **Importance of SDLC:** Software development methods; **Web application security principles:** Application design & development security, Environment and controls, Essence of secure software development Introduction to input validation & sensitive data: Implementation of input validation, Practical solutions; **Input validation vulnerability:** Buffer overflow, Cross site scripting, SQL injection, Canonicalization, Sensitive data, Sensitive data access, Sensitive data in storage, Information disclosure, Data tampering.

Unit II

Introduction to authentication & authorization: Network eavesdropping, Brute force attack, Dictionary attack, Cookie replay attack, Credential theft, Elevation of privilege, Basic of authorization, Fata tampering, Luring attack, Phishing attack. **Introduction to configuration management & session management:** Unauthorized access to administration interfaces, Unauthorized access to configuration stores, Retrieval of clear text configuration data, Lack of individual accountability, Over-privileged process and services accounts; Basics of session management: Hijacking attack, Session replay attack, Man in the middle attack.

Unit III

Introduction to cryptography, parameter manipulation & exception management: Introduction, Poor key generation or key management, Weak or custom encryption, Basics of parameter manipulation, Cookie manipulation, HTTP Header manipulation, Basics of exception management, Denial of services. Auditing & logging, Countermeasures: Introduction to auditing & logging, countermeasures, Basic countermeasures.

Unit IV

Data security threats: IT Data security, Need of data security, Importance of data security, Critical data for organization, Elements to consider for a better security mechanisms, Types of data security threats: Malware threat, Network based threats, Cryptographic threats, Database security threats, Types of data security threats: Banking fraud threats, Web application threats, Physical security threats, Wireless network security threats, Bluetooth devices threats, Data threats in modern era, Benefits of data security. **Data security threat techniques:** Introduction, Threat techniques, Network based threat techniques, Cryptographic threat techniques, Banking fraud techniques, Web-application threat techniques, Wireless network threat techniques**.**

Unit V

Countermeasures: IT data security – countermeasures, The importance of data protection, Evolution of mitigation techniques, Countermeasures, application layer protocol

Text Book:

1 Data and Application Security ,Developments and Directions Bhavani Thuraisingham, Reind Riet, Klaus R. Dittrich, Zahir Tari

2 Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes,Computer Forensics and Legal Perspectives,Wiley

3 B.B.Gupta, D.P.Agrawal,HaoxiangWang,ComputerandCyberSecurity:Principle s, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335,2018

Course Code	**LMCA:104**
Course Title	**Programming fundamentals using Python**
Credits	4
Course Objective(s)	Upon completing the "Programming Fundamentals Using Python" course, students will be able to write efficient Python code, solve problems using algorithms and data structures, debug programs, and develop simple applications. This foundational knowledge will prepare them for advanced studies in computer science and software development.
Course Outcome (CO)	**CO1: Understand Basic Concepts**: Introduce students to fundamental programming concepts such as variables, data types, control structures, and functions. **CO2**: **Develop Problem-Solving Skills:** Equip students with the ability to analyze problems and design algorithms to solve them using Python. **CO3**: **Implement Security Measures:** Learn to apply security measures and best practices to protect IT applications and data from unauthorized access and breaches. **CO4**: **Practice Debugging Techniques:** Enable students to identify and fix errors in their code through effective debugging techniques. **CO5: Work with Libraries and Modules:** Introduce students to Python libraries and modules, encouraging code reuse and enhancing functionality.

LMCA: 104 Programming fundamentals using Python

Unit I

Introduction to Python Language: Overview, Features of Python, Execution of a Python Program, Innards of Python, Frozen Binaries, Python Interpreter, Comparison of Python with C and Java, Installing Python, Writing & Executing, IDLE Data Types, Variables And Other Basic Elements: Comments, Docstrings, Data types- Numeric, Compound, Boolean, Dictionary, Sets, Mapping, Basic Elements of Python, Variables Input and Output Operations: Input Function, Output Statements, Command Line Arguments Control Statements: Control Statements- Loop Statement, The else Suite, break Statement, continue Statement, pass Statement, assert Statement, return Statement

Unit II

Functions: Defining & Calling a Function, Returning Results, Returning Multiple Values, Built-in Functions, Parameters and Arguments, Recursive Functions, Anonymous or Lambda Functions Operators: Arithmetic operators, Assignment operators, Unary minus operator, Relational operators, Logical operators, Bitwise operators, Membership operators, Identity operators, Precedence of Operators, Associativity of Operators Arrays: Creating Arrays, Indexing and Slicing, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic Slicing. Advanced Indexing. Dimensions of Arrays, Attributes of an Array

Strings: Creating Strings, Functions of Strings, Working with Strings, Length of a String, Indexing & Slicing, Repeating & Concatenation of Strings, Checking Membership, Comparing Strings, Removing Spaces, Finding Substrings, Counting Substrings, Strings are Immutable, Splitting and Joining Strings, Changing Case, Checking Starting and Ending of a String, Sorting & Searching in the Strings, Formatting the Strings, Working with Characters

Unit III

Lists and Tuples: Lists, List Functions and Methods, List Operations, Tuples Dictionaries: Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries, Ordered Dictionaries Regular Expressions: What is a Regular Expression? Sequence Characters in Regular Expressions, Quantifiers in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expression on Files, Retrieving Information from an HTML File Date and Time in Python: Date and Time, Date and Time Now, Combining Date and Time, Formatting Dates and Times, Finding Durations using “timedelta”, Comparing Two Dates, Sorting Dates, Stopping Execution Temporarily, Knowing the Time taken by a Program, Working with Calendar Module

Unit IV

IPython: Beyond Normal Python, Help and Documentation in IPython, Keyboard Shortcuts in the IPython Shell, IPython Magic Commands, Input and Output History, IPython and Shell Commands, Errors and Debugging, Profiling and Timing Code Introduction to NumPy: Understanding Data Types in Python, The Basics of NumPy Arrays, Computation on NumPy Arrays: Universal Functions, Aggregations: Min, Max, and Everything In Between, Computation on Arrays: Broadcasting, Comparisons, Masks, and Boolean Logic, Fancy Indexing, Sorting Arrays, Structured Data: NumPy's Structured Arrays

Unit V

Data Manipulation with Pandas: Introducing Pandas Objects, Data Indexing and Selection, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concat and Append, Combining Datasets: Merge and Join, Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series. High-Performance Pandas: eval( ) and query( ) Visualization with Matplotlib: Simple Line Plots, Simple Scatter Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binnings, and Density, Customizing Plot Legends, Customizing Colorbars, Multiple Subplots, Text and Annotation, Customizing Ticks, Customizing Matplotlib: Configurations and Stylesheets, Three-Dimensional Plotting in Matplotlib, Geographic Data with Basemap, Visualization with Seaborne

Text Books

1. Fundamentals of Python Programming Richard L. Halterman Southern Adventist University (2019), and Internet Archive
2. THINK PYTHON: HOW TO THINK LIKE A COMPUTER SCIENTIST by Allen B. Downey

Course Code	**LMCA: 105**
Course Title	**Programming Through C++**
Credits	4
Course Objective(s)	The **Programming Through C++ course** equips students with the skills to write efficient programs using object-oriented principles like encapsulation, inheritance, and polymorphism. Students learn to solve problems using C++ constructs, algorithms, and data structures, while mastering advanced features such as dynamic memory allocation, file handling, and the Standard Template Library (STL). The course emphasizes practical application through projects, preparing learners for advanced programming roles and software development challenges.
Course Outcome (CO)	**CO1: Understand the Fundamentals of Programming:** Learn basic programming concepts such as variables, data types, operators, control structures, and functions. **CO2: Master Object-Oriented Programming Concepts:** Gain proficiency in OOP principles, including classes, objects, inheritance, polymorphism, encapsulation, and abstraction. **CO3: Develop Problem-Solving Skills:** Apply structured and modular approaches to solve computational problems using C++. **CO4: Work with Advanced C++ Features:** Understand pointers, memory management, file handling, templates, and exception handling for efficient programming. **CO5: Enhance Debugging and Testing Abilities** Use debugging tools and techniques to write error-free and optimized programs. **CO6: Practical Application and Software Development:** Design and implement real-world applications, reinforcing concepts through hands-on experience and projects. **CO7: Prepare for Advanced Computing Courses:** Build a foundation for exploring advanced programming, algorithms, and software engineering concepts in future courses.

LMCA: 105 Programming through C++

UNIT-I

OOP, need for OOP, differences between OOP and procedure oriented programming, abstraction, overview of OOP principles-, class, object, en-capsulation, inheritance and data binding, polymorphism, abstraction.

C++basics: structure of a C++program, data types, declaration of variables, expressions, operators, Type conversions, Introduction to C++ Stream, flow control statement, functions, Classification of Functions, parameter passing, recursive functions, default arguments, inline functions, Arrays & Strings, Structures.

UNIT–II

C++classes and data abstraction: class definition, class structure, class objects, class scope, this pointer, static class members ,constant member functions, constructors and destructors, dynamic creation and destruction of objects, friend function and class, static class member.
Overloading: function overloading, operator overloading unary, binary operators.

UNIT-III

Inheritance: defining a class hierarchy, different forms of inheritance, defining the base and derived classes, access to the base class members, base and derived class construction, destructors, virtual base class.

UNIT–IV

Polymorphism: static and dynamic bindings, base and derived class virtual functions, dynamic binding through virtual functions, virtual function call mechanism, pure virtual functions, abstract classes.

Exception handling: benefit so exception handling, throwing an exception, the try block, catch exception.

UNIT-V

Templates-function templates and class templates
File handling: stream classes hierarchy, stream I/O, file streams, opening and closing data file, creating a data file, read and write file, error handling during file operation.

Recommended Books:

1. Bjarne Stroustrup, “C++ Programming Language”, Addison Wesley
2. Balagurusamy E, “Object Oriented Programming with C++”, TMH, 2001
3. Booch Grady, “Object Oriented Analysis and Design with application 3/e”, Pearson
4. Lipman, Stanley B, JonsceLajole, “ C++ Primer Reading”, AWL, 1999
5. Horowitz and Sahani, “Fundamentals of data Structures”, Galgotia

Course Code	**LMCA: 106**
Course Title	**Web Programming**
Credits	4
Course Objective(s)	The Web Programming course equips students with essential skills in modern web development. Students will master HTML, CSS, and JavaScript for creating responsive web applications, and gain experience in front-end frameworks like React and back-end technologies such as PHP or Node.js. The course also covers database management, web security, and accessibility, ensuring the development of user-friendly applications. Through collaborative projects, students will enhance their teamwork skills and build a portfolio to showcase their expertise, preparing them for successful careers in the tech industry.
Course Outcome (CO)	**CO1: Understanding Web Technologies**: Gain a comprehensive understanding of the fundamental technologies used in web development, including HTML, CSS, JavaScript, and server-side programming languages.. **CO2: Front-End Development Skills:** Develop skills in creating responsive and user-friendly web interfaces using modern front-end frameworks and libraries (e.g., React, Angular, or Vue.js). **CO3**: **Back-End Development Skills:** Learn to build server-side applications using languages such as PHP, Python, Ruby, or Node.js, and understand how to interact with databases (e.g., MySQL, MongoDB). **CO4: Database Management:** Understand the principles of database design and management, including SQL and NoSQL databases, and how to integrate them into web applications. **CO5**: **Web Application Architecture:** Explore different architectural patterns for web applications, including MVC (Model-View-Controller) and Restful services, and understand how to design scalable and maintainable applications. **CO6**: **Version Control and Collaboration:** Learn to use version control systems like Git for collaborative development and project management.

LMCA: 106 Web Programming

UNIT-I

The internet: World wide web, web browser, Web Server.
Introduction of HTML: introduction, mark up language, editing HTML, common tags, headers, text styles, linking, images, formatting text, horizontal rules and more line breaks, unordered lists, nested and ordered lists, basic HTML tables.

UNIT-II

Formatting: basic HTML forms, HTML frames, internal linking, image maps.
Dynamic HTML : CSS : introduction – inline styles, creating style sheets with the style element, conflicting styles, linking external style sheets, backgrounds, element dimensions.

UNIT-III

Dynamic HTML: object model and collections: introduction, Dynamic style, Dynamic Positioning.
Dynamic HTML: event model : introduction, event ON CLICK, event ON LOAD

UNIT-IV

Java script – introduction to scripting, Advantages of java script: introduction-java script in head & body section, memory concepts- Operators- decision making. Java script control structures.
Java script functions: introduction function definitions, duration of identifiers, scope rules, java script global functions.

UNIT-V

Java script arrays: introduction, array-declaring and allocating arrays
Java script objects: introduction, math, string, data, Boolean and number objects.
Introduction to PHP – Advantages of PHP – Data types

Recommended Books:

1. ShishirGundavarma, “CGI Programming on the World Wide Web”, O’Reilly & Associate.
2. DON Box, “Essential COM”, and Addison Wesley.
3. Burdman, “Collaborative Web Development”, Addison Wesley.
4. Sharma &Sharma, “Developing E-Commerce Sites”, Addison Wesley
5. Ivan Bayross, “Web Technologies Part II”, BPB Publications.

Course Code	**LMCA: 107**
Course Title	**Programming Through C++ Lab**
Credits	2
Course Objective(s)	Upon completing the Programming Through C++ Lab course, students will have gained proficiency in C++ syntax and object-oriented programming principles, enabling them to write efficient and robust code. They will develop strong problem-solving skills to tackle complex programming challenges and gain experience in debugging and testing their applications. Additionally, students will understand fundamental data structures and algorithms, enhancing their ability to manage data effectively. Through collaborative projects, they will improve their teamwork and communication skills while creating practical applications. Overall, the course will prepare students for advanced studies in programming and successful careers in software development.
Course Outcome (CO)	**CO1: Understand Fundamental Concepts: Familiarize students with basic** programming concepts such as data types, control structures, functions, and arrays. . **CO2: Object-Oriented Programming:** Introduce students to object-oriented programming principles, including classes, objects, inheritance, polymorphism, and encapsulation. **CO3: Problem-Solving Skills:** Enhance students' problem-solving abilities by engaging them in practical coding exercises and projects that require logical thinking and algorithmic design. **CO4: Debugging and Testing:** Teach students effective debugging and testing techniques to identify and resolve errors in their code. **CO5: Real-World Applications**: Encourage the application of C++ programming skills to solve real-world problems through projects and case studies. **CO6: Software Security Standards:** Familiarize with industry standards and best practices for secure software development.

List of Lab Programs:

1. Write a program that calculates the average grade of a class of students based on user input.

2. Write a program that checks if a given string is a palindrome, ignoring spaces, punctuation, and case.

3. Create a class Bank Account that allows users to create an account, deposit money, withdraw money, and check their balance.

4. Implement a bubble sort algorithm to sort an array of integers input by the user.

5. Write a program that reads a text file and counts the number of words in it, prompting the user for the filename.

6. Write a program that converts temperatures between Celsius and Fahrenheit based on user input.

7. Create a simple console-based Tic-Tac-Toe game for two players.

8. Write a program that multiplies two matrices input by the user and displays the result.

9. Write a program that checks if a given number is prime.

10. Create a program that allows users to add, view, and delete contacts from a simple contact list.

Course Code	**LMCA: 108**
Course Title	**Web Programming Lab**
Credits	2
Course Objective(s)	The Web Programming Lab course aims to equip students with essential web development skills, covering core technologies such as HTML, CSS, and JavaScript. By the end of the course, students will be proficient in creating responsive web designs, developing dynamic applications through client-side and server-side programming, and interacting with databases. They will also learn best practices for web security, utilize version control with Git, and deploy applications to live servers. Emphasizing user experience design and project management, the course prepares students for entry-level positions in web development or further studies, fostering critical thinking and problem-solving skills in the rapidly evolving web technology landscape.
Course Outcome (CO)	**CO1: Understanding Web Technologies:** Gain a foundational understanding of the core technologies used in web development, including HTML, CSS, and JavaScript. **CO2: Responsive Design Principles:** Learn to create responsive web designs that provide an optimal viewing experience across a wide range of devices, including desktops, tablets, and smart phones. **CO3: Client-Side Scripting:** Develop proficiency in client-side scripting using JavaScript to enhance user interaction and create dynamic web pages. **CO4: Server-Side Programming:** Understand the basics of server-side programming and how to interact with databases using languages such as PHP, Node.js, or Python. **CO5: Web Application Development**: Gain hands-on experience in developing full-fledged web applications, including front-end and back-end components. . **CO6: Database Management:** Understand the basics of database management.

List of Lab Programs:

1. Create a simple web page with a header, a paragraph, and an image.
2. Write CSS to style a web page with a specific color scheme and layout.
3. Modify a web page to make it responsive using media queries.
4. Create a CSS animation that changes the background color of a button on hover.
5. Write a function to change the text of an HTML element when a button is clicked.
6. Create an HTML form and validate the input fields using JavaScript.
7. Write a program that displays the pressed key on the web page when a key is pressed.
8. Make an AJAX request to a public API and display the fetched data.
9. Create a web application that stores a user's name in local storage and displays it on page refresh.
10. Write a simple Node.js server that responds with "Hello, World!".
11. Create a RESTful API using Express.js for a to-do list with CRUD operations.
12. Write a script to connect to a MongoDB database and retrieve documents.
13. Implement a user authentication system using JSON Web Tokens (JWT).
14. Write a server-side script to handle file uploads.
15. Write a function to sort an array of numbers in ascending order.
16. Create a function that generates the Fibonacci sequence up to a specified number.
17. Write a function to check if a string is a palindrome.
18. Build a simple calculator that performs basic arithmetic operations.

Course Code	**LMCA: 109**
Course Title	**Programming Fundamentals using python Lab**
Credits	2
Course Objective(s)	Upon completing the **Programming Fundamentals using Python Lab** course, students will have developed strong programming skills in Python. They will be able to write, test, and debug Python code, implementing basic algorithms and solving problems efficiently. Students will gain hands-on experience with key programming concepts, including control structures, functions, data structures, and file handling, enabling them to approach real-world challenges with a solid foundation in Python programming.
Course Outcome (CO)	**CO1: Learn Python Syntax and Basic Concepts**: Understand basic programming concepts like variables, data types, and operators. **CO2: Master Control Structures**: Implement decision-making structures (if, else) and loops (for, while). **CO3: Develop Problem-Solving Skills**: Practice breaking down problems into smaller, manageable tasks and writing Python code to solve them. **CO4**: Understand **Functions and Modular Programming**: Learn how to define and use functions to organize code and promote reusability. **CO5: Work with Collections**: Use Python data structures such as lists, tuples, and dictionaries to store and manipulate data. **CO6: Debugging and Error Handling**: Gain experience in identifying and fixing bugs, and handle errors with proper exception handling.

List of Lab Programs:

1. Write a Python program to input a number and check if it is positive, negative, or zero.
2. Write a Python program to find the largest of three numbers entered by the user.
3. Write a Python program that takes a list of numbers and prints the sum and average of the list.
4. Write a Python program to count the number of vowels in a given string.
5. Write a Python program to reverse a given string.
6. Write a Python program that reads a text file, counts the number of words, and displays them.
7. Write a Python program to check if a given number is a prime number.
8. Write a Python program to implement a basic calculator that can perform addition, subtraction, multiplication, and division.
9. Write a Python program that accepts a list of integers and prints all even numbers from the list.
10. Write a Python program that takes a list of strings and prints them in alphabetical order.
11. Write a Python program to create a simple to-do list, where the user can add, remove, or view tasks.
12. Write a Python program to read a text file, count the number of occurrences of a specific word, and display it.
13. Write a Python program to implement a basic ATM machine simulation, where the user can check balance, deposit, and withdraw money.
14. Write a Python program that finds the factorial of a given number using a loop.
15. Write a Python program to merge two lists into a single list without duplicates.

Course Code	**LMCA-110**
Course Title	**Seminar**
Credits	2
Course Objective(s)	Upon completion of the seminar course, participants will have gained a thorough understanding of key concepts and contemporary issues related to the topic. They will demonstrate improved critical thinking, research, and communication skills, and will be able to apply theoretical knowledge to real-world scenarios. Additionally, participants will enhance their ability to collaborate and network with peers, preparing them for future academic and professional pursuits.
Course Outcome (CO)	**CO1: Knowledge Acquisition:** To equip participants with a deep understanding of the seminar topic, including foundational theories, key concepts, and current trends. **CO2: Critical Thinking:** To develop participants' ability to analyze, evaluate, and synthesize information from various sources, fostering independent thought and critical reasoning. **CO3: Research Skills:** To enhance participants' research capabilities by engaging them in literature reviews, data analysis, and the exploration of relevant case studies. **CO4: Effective Communication:** To improve participants' oral and written communication skills through presentations, discussions, and the preparation of seminar papers or reports**.** **CO5: Collaboration:** To promote teamwork and collaborative learning by involving participants in group projects, discussions, and peer evaluations.

PAHER University, Udaipur
Faculty of Computer Science
LMCA (Master of Computer Applications)- 2Yr Course
SEMESTER-II

Course Code	**LMCA: 201**
Course Title	**Advanced Database Management Systems-II**
Credits	4
Course Objective(s)	The Advanced Database Management Systems (ADBMS) course provides students with a deep understanding of complex database concepts, including distributed databases, data warehousing, and big data technologies. Students will learn to design intricate database schemas, optimize SQL queries for performance, and manage transactions effectively.. Through practical projects and case studies, learners will enhance their problem-solving skills and apply theoretical knowledge to real-world database management challenges.
Course Outcome (CO)	**CO1**: **Deepen Knowledge:** Enhance students' understanding of advanced database models, including object-based databases, distributed databases, and parallel databases, and their applications in real-world scenarios. **CO2: Develop Practical Skills:** Provide hands-on experience in using PL/SQL for database programming, including the creation of stored procedures, functions, and the effective use of cursors for data manipulation**.** **CO3: Foster Analytical Thinking:** Encourage students to analyze complex database problems, design efficient database architectures, and implement optimization techniques for performance enhancement**..** **CO4**: **Promote Collaboration and Research**: Facilitate collaborative learning through group projects and case studies, allowing students to engage with current research and industry practices in advanced database management**.**

LMCA: 201 Advanced Database Management Systems-II

UNIT-I

Object Base Database: -Object –Oriented data model, Object Oriented Language, Résistance Programming Object Relation Database: Nested Relation, Complex type, Inheritance, Refrence type, query with complex type's function and procedure storage for object database

UNIT-II

Distributed Database:-Distributed data storage, Distributed Transactions, Commit protocol, concurrency control in distributed database, availability, distributed query processing.

UNIT-III

Parallel Database:-I/O parallelism, interquery parallelism, intraquery parallelism, interoperation parallelism, interoperation parallelism, design of parallelism

UNIT IV

PL/SQL basics, Comparison between SQL and PL/SQL, Advantages & Disadvantages of PL/SQL, blocks, architecture, variables an constants, attributes, character set, PL/SQL sentence structure, data types, conditional control statements, control structures.

UNIT V

Cursors, exceptions, triggers, procedures and packages

RecommendedBooks:

1. Majumdar& Bhattacharya, "Database Management System", TMH.
2. Korth, Silbertz, Sudarshan, " Database Concepts", McGraw Hill.
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Addison Wesley.
4. Data C J," An Introduction To Database System", Addison Wesley.
5. Ramakrishnan, Gehrke, "Database Management System", McGraw Hill.

Course Code	**LMCA: 202**
Course Title	**.Net Framework and Programming in asp.net**
Credits	4
Course Objective(s)	Completing the .NET Framework and ASP.NET course enables students to design, develop, and deploy secure, scalable web applications using Microsoft's .NET Framework and ASP.NET. They will master object-oriented programming, data access techniques, security, state management, exception handling, web services, and performance optimization, preparing them for successful careers in software development..
Course Outcome (CO)	**CO1**: **Introduction to .NET Framework:** Understanding the fundamentals of the .NET Framework, including its architecture, common language runtime (CLR), and base class libraries.. **CO2: ASP.NET Basics:** Learning the basics of ASP.NET, including creating web pages, handling user input, and managing application state. **CO3: Object-Oriented Programming:** Mastering object-oriented programming concepts, such as inheritance, encapsulation, and polymorphism, within the context of the C# programming language. **CO4: ASP.NET Controls:** Utilizing ASP.NET controls, such as web forms, validation controls, and data-bound controls, to create interactive and dynamic web pages. **CO5**: **Data Access:** Implementing data access techniques using ADO.NET, Entity Framework, and LINQ to interact with databases and external data sources**.** **CO6**: **Security**: Understanding and implementing security measures, including user authentication, authorization, and input validation, to protect web applications from potential threats. **CO7**: **Exception Handling:** Implementing robust exception handling techniques to ensure applications can gracefully handle unexpected errors and conditions**.**

LMCA: 202 .Net Framework and Programming in asp.net

Unit -I

Introduction to .NET Framework : Genesis of .Net – Features of .Net - .Net binaries – Microsoft Intermediate Language – Meta Data - .Net types and .net name spaces – Common Language Runtime – Common Type System – Common Language Specification - .Net Applications using command line compiler and visual studio .net IDE.

Unit -II

Basics of ASP.NET : Introducing ASP .NET – Creating and deploying ASP .NET applications – Web forms – Web controls – working with events – Rich web controls – Custom web controls – Validation controls – Debugging ASP .NET pages.

Unit -III

Advanced ASP .NET : ASP .NET configuration – Business objects – HTTP Handlers – Caching in ASP .NET – ASP .NET security – Localizing ASP .NET applications – Deployment projects.

Unit -IV

Building Web Services : Introduction to web services – Web services Infrastructure – SOAP – Building a web service – Deploying and publishing web services – Finding web services – Consuming web services.

Unit -V

Ado .NET: Basics of ADO .NET – Changes from ADO – Data Table – Data Views – Data Set – Data Relation Type – ADO .NET Managed Providers – OLEDB and SQL Managed Providers – OleDb Data Adapter Type.

RecommendedBooks:

1. ASP.NET Black Book by stevenholzner –dreamtech
2. ASP.NET Unleashed
3. C# programming – wrox publication
4. C# programming Black Book by Matt tells

Course Code	**LMCA: 203**
Course Title	**Advanced JAVA-II**
Credits	4
Course Objective(s)	Upon completing the Advanced Java-II course, students will gain a strong understanding of advanced Java concepts and technologies. They will be proficient in using frameworks like Spring and Hibernate for enterprise application development, apply design patterns for better software architecture, and manage multithreading and concurrency effectively. Students will learn to design and consume RESTful and SOAP web services, handle database management with JDBC and ORM frameworks, and implement security best practices. Additionally, they will develop skills in testing and debugging with tools like JUnit and Mockito, understand microservices architecture, and optimize application performance. Through hands-on projects, students will be well-prepared to address complex software development challenges in the industry..
Course Outcome (CO)	**CO1**: **Mastering Java Frameworks:** Gain proficiency in popular Java frameworks such as Spring and Hibernate, enabling students to build robust and scalable enterprise applications. **CO2: Understanding Design Patterns:** Learn and apply common design patterns to solve software design problems, promoting best practices in software architecture and development. **CO3: Concurrency and Multithreading:** Explore advanced multithreading and concurrency concepts, allowing students to develop applications that efficiently utilize system resources. **CO4: Web Services and APIs:** Understand how to create and consume RESTful and SOAP web services, facilitating communication between distributed systems. **CO5**: **Database Connectivity:** Enhance skills in database management and connectivity using JDBC and ORM frameworks, ensuring effective data handling in applications. **CO6**: **Java Security**: Learn about security best practices in Java applications, including authentication, authorization, and secure coding techniques. **CO7**: **Testing and Debugging**: Develop skills in unit testing and debugging Java applications using tools like JUnit and Mockito, ensuring code quality and reliability.

LMCA: 203 Advanced JAVA-II

Unit -I

J2EE Platform: Enterprise architecture style: 2 tier, 3 tier, n tier, J2EE platform: run time APIs, J2EE technology: web components, EJB.

Unit - II

Programming Fundamentals Data Types, Variables, Parse, Control Structures, Decision Structures: if, if-else, switch, Looping Structures: while, do-while, for Accessing MySQL database JDBC to connect with database

Unit -III

JavaScript :- Basic Introduction of HTML, Introducing of Java Script, Basic Syntax data type variables, operator, conditional statement & looping statement, writing function, form validation.

Unit -IV

Database Programming with JDBC: Database drivers, java.sql package, javax.sql package, connectionpooling,Basic step of JDBC.
Servlet Programming: Servlet implementation, Servlet configuration, Servlet life cycle. Accessing database, Servlet chaining, session tracking Management, Get & Post request.

Unit -V

JSP Overview ,JSP Architecture ,JSP – Life Cycle,JSPDeclarations, JSPDirectives, JSPActions, JSP Implicit Objects, page Directive ,include Directive, JSP Actions, request Object, The response Object, out Object, GET Method, POST Method , session Management , JSP With Database

Recommended Books:

1. W. Jason Gilmore: Beginning PHP and MySQL, Apress, New Delhi-USA.
2. P.J.Deitel& H. M. Deitel: Internet & World Wide Web, How to program, PHI New Delhi.
3. The complete reference of J2EE, Tata McGraw Hill Publishing Company New Delhi.
4. Complete J2EE, Black Book

Course Code	**LMCA: 204**
Course Title	**Internet Of Things**
Credits	4
Course Objective(s)	Upon completing the IoT course, students will gain the skills to design, implement, and manage IoT systems, including working with sensors, networks, and cloud platforms. They will be prepared to develop IoT solutions, address security and scalability challenges, and apply their knowledge to real-world applications across various industries.
Course Outcome (CO)	**CO1**: **Understanding IoT Fundamentals:** Define the Internet of Things and its significance in modern technology. Explain the key components of IoT systems, including sensors, actuators, connectivity, and data processing. **CO2**: **Knowledge of IoT Architecture**: **Study the layers of IoT architecture**, including perception, network, and application layers. **Explore the communication protocols** used in IoT, such as MQTT, CoAP, HTTP, and Bluetooth. **CO3**: **IoT Devices and Sensors**: Learn about IoT hardware, including microcontrollers, sensors, actuators, and embedded systems. Understand sensor data acquisition, processing, and transmission. **CO4**: **Network Connectivity and Communication: Examine the networking protocols** specific to IoT, such as Zigbee, LoRa, NB-IoT, and 5G. **Understand challenges in IoT communication**, including bandwidth limitations, energy consumption, and security. **CO5**: **IoT Platforms and Cloud Computing**: **Get hands-on experience with IoT platforms** for device management, data collection, and application deployment. **Explore cloud computing integration** for data storage, analytics, and scalability in IoT systems.

LMCA: 204 INTERNETS OF THINGS

UNIT-I

Fundamentals of IoT: Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.

UNIT-II

Sensors Networks : Definition, Types of Sensors, Types of Actuators, Examples and Working, IoT Development Boards: Arduino IDE and Board Types, RaspberriPi Development Kit, RFID Principles and components, Wireless Sensor Networks: History and Context, The node, Connecting nodes, Networking Nodes, WSN and IoT.

UNIT-III

Wireless Technologies for IoT: WPAN Technologies for IoT: IEEE 802.15.4, Zigbee, HART, NFC, Z-Wave, BLE, Bacnet, Modbus. IP Based Protocols for IoT IPv6, 6LowPAN, RPL, REST, AMPQ, CoAP, MQTT. Edge connectivity and protocols

UNIT-IV

Data Handling& Analytics: Introduction, Bigdata, Types of data, Characteristics of Big data, Data handling Technologies, Flow of data, Data acquisition, Data Storage, Introduction to Hadoop. Introduction to data Analytics, Types of Data analytics, Local Analytics, Cloud analytics and applications

UNIT-V

Applications of IoT: Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.

Text Books:

1. Hakima Chaouchi, — “The Internet of Things Connecting Objects to the Web” ISBN : 978-1- 84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, — “The Internet of Things: Key Applications and Protocols”, WileyPublications
3. Vijay Madisetti and ArshdeepBahga, — “Internet of Things (A Hands-on-Approach)”, 1 st Edition, VPT, 2014.
4. J. Biron and J. Follett, "Foundational Elements of an IoT Solution", O'Reilly Media, 2016.
5. Keysight Technologies, “The Internet of Things: Enabling Technologies and Solutions for Design and Test”, Application Note, 2016.

Course Code	**LMCA: 205**
Course Title	**Data Structures Using C++**
Credits	4
Course Objective(s)	Upon completing the Data Structures Using C++ course, students will gain a solid understanding of various data structures, including arrays, linked lists, stacks, queues, trees, and graphs, and their applications. They will be proficient in implementing these structures in C++ and analyzing the efficiency of associated algorithms using Big O notation. Students will learn to select and apply appropriate data structures to solve computational problems, as well as design and optimize algorithms for data manipulation. Additionally, they will develop debugging skills and collaborate on projects, enhancing their teamwork and communication abilities. This course will provide a strong foundation for further studies in algorithms and computer science, equipping students to tackle complex programming challenges effectively.
Course Outcome (CO)	**CO1**: **Fundamental Concepts:** Understand the fundamental concepts of data structures, including their definitions, types, and applications in problem-solving. **CO2**: **Implementation Skills:** Develop the ability to implement various data structures (such as arrays, linked lists, stacks, queues, trees, and graphs) using C++ programming language. **CO3: Algorithm Development:** Learn to design and analyze algorithms for manipulating data structures, focusing on efficiency in terms of time and space complexity. **CO4**: **Application of Data Structures**: Apply appropriate data structures to solve real-world problems, demonstrating the ability to choose the right structure based on specific requirements. **CO5**:**Advanced Data Structures:** Explore advanced data structures such as hash tables, heaps, and balanced trees, understanding their use cases and implementation techniques.

LMCA: 205 Data Structures Using C++

UNIT-I

Arrays, records, stack, operation on stack, implementation of stack kasanarray, queue, operations on queue, implementation of queue, application of queue**.**

UNIT-II

Linked list: List representation, operations on linked list-get node and free node operation, implementing the list operation, insertingintoan ordered linked list, deleting, circular linked list, doubly linked list.

UNIT-III

Tree Structure: Binary search tree, inserting, deleting and searching into binary search tree, implementing the insert, search and delete algorithms, tree traversals, B + tree and AVL Tree

UNIT-IV

GraphStructure: Graphrepresentation-Adjacencymatrix,adjacencylist. Orthogonalrepresentationofgraph.Graphtraversalsbfsanddfs.Shortestpath, all pairs of shortest paths,

UNIT-V

Searching and sorting: Searching-sequential searching, binary searching, hashing. Sorting selection sort, bubble sort, quick sort, heap sort, merge sort, and insertion sort,

Recommended Books

1. HorowitzESartajSahni,FundamentalsofDataStructure, GalgotiaPublicationPrivateLimited., New Delhi.
2. R.S. Saleria, Data Structure and algorithm using C.
3. R.B. Patel, Expert data structure with C.

Course Code	**LMCA: 206**
Course Title	**Cloud Computing**
Credits	4
Course Objective(s)	Upon completing the Cloud Computing course, students will gain a strong understanding of cloud models, architectures, and service platforms like AWS, Azure, and Google Cloud. They will be able to deploy, manage, and scale cloud resources, implement security measures, and optimize cloud solutions for cost and performance. Additionally, students will be prepared to design and deploy scalable applications while staying up-to-date with emerging cloud technologies.
Course Outcome (CO)	**CO1**: **Understanding Cloud Computing Fundamentals**: Learn the core concepts of cloud computing, including virtualization, cloud deployment models (private, public, hybrid), and service models (IaaS, PaaS, SaaS). **CO2**: **Exploring Cloud Architecture**: Study cloud architecture components such as servers, storage, databases, networking, and security mechanisms.. **CO3: Cloud Service Providers**: Gain hands-on experience with major cloud platforms like AWS, Microsoft Azure, and Google Cloud, and learn how to deploy and manage resources on these platforms. **CO4**: **Cloud Storage and Databases**: Understand cloud-based storage solutions, database management, and data retrieval techniques in a cloud environment. **CO5Security and Privacy in the Cloud**: Explore cloud security challenges and solutions, including identity management, encryption, and compliance..

LMCA: 206 Cloud Computing

Unit I

Cloud Computing Overview, Origins of Cloud computing, Cloud components, Characteristics of cloud computing, Benefits and Challenges of Cloud Computing, Cloud Computing Planning, Cloud Computing Architecture.

Unit II

Cloud Computing Models: Deployment Model, Public Cloud, Private Cloud, Community Cloud, Hybrid Cloud, Service Model Infrastructure as a Service (IaaS): IaaS service providers – Amazon AWS– Benefits, Platform as a Service (PaaS): PaaS service providers – Salesforce.com – Services and Benefits, Software as a Service (SaaS): SaaS service providers – Google App Engine, Salesforce.com– Benefits

Unit III

Virtualization: Virtual Machine, Virtualization in Cloud, Hardware Virtualization, Types of hardware virtualization: Full virtualization - Emulation virtualization - para virtualization, Desktop Virtualization, Introduction to other Virtualization, Benefits, Introduction to Cloud Computing Applications**:** Business Applications, Management Applications, Social Applications, Entertainment Applications, Art Applications, Cloud Computing Providers.

Unit IV

Data Storage, Types of Storage, Backup, Types of backup (Full, Incremental, Differential, Local, Off Line, Online, Remote, Cloud Back Up)

RAID (Redundant Array of Inexpensive Disks) Technology, Types of RAID, RAID Levels, RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5, RAID 6,Mirroring, Data Striping, Advantages and Disadvantages

Unit V

Understanding cloud based security issues and threats, SQL query injections, SSL, Authentication and identity.

Recommended Books:

1. PrasantaPattnaik, ManasKadat, Souvik Pal, Fundamentals of Cloud Computing.

2. Thomas erl, Cloud computing concepts, Technology & Architecture

3.Dr. Kumar Saurabh, Cloud Computing

Course Code	**LMCA: 207**
Course Title	**Advanced Database Management System-II Lab**
Credits	2
Course Objective(s)	Upon completing the Advanced Database Management System-II Lab course, students will be proficient in writing complex SQL queries, designing and normalizing database schemas, and managing transactions to ensure data integrity. They will have hands-on experience with stored procedures, triggers, and NoSQL databases, as well as an understanding of data warehousing and mining techniques. Students will also learn to optimize database performance and collaborate effectively on projects, preparing them for careers in database administration, data analysis, and software development. Overall, they will be equipped with the skills necessary to tackle real-world database challenges.
Course Outcome (CO)	**CO1**: **Hands-On Experience:** Provide students with practical experience in using advanced database management systems, enabling them to apply theoretical concepts learned in lectures to real-world scenarios**.** **CO2**: **SQL Proficiency**: Enhance students' skills in writing complex SQL queries, including joins, subqueries, and set operations, to manipulate and retrieve data effectively from relational databases**.** **CO3: Database Design:** Enable students to design and implement advanced database schemas, including normalization techniques and entity-relationship modeling, to ensure efficient data organization. **CO4**: **Transaction Management:** Familiarize students with transaction management concepts, including ACID properties, concurrency control, and recovery techniques, to maintain data integrity in multi-user environments. **CO5**: **Stored Procedures and Triggers**: Teach students how to create and utilize stored procedures, functions, and triggers to automate database operations and enforce business rules.

List of Lab Programs:

1. Write a SQL query to retrieve all columns from a table named Employees where the Department is 'Sales'.
2. Write a SQL query to find the names of employees and their corresponding department names by joining the Employees and Departments tables.
3. Write a SQL query to find all employees who earn more than the average salary of their department.
4. Write a SQL query to count the number of employees in each department and display the results in descending order of count.
5. Write a SQL query to find the employees who are either in the 'Sales' department or have a salary greater than $70,000.
6. Given a table structure, identify any normalization issues and redesign the schema to achieve at least 3NF.
7. Create an Entity-Relationship diagram for a library management system that includes entities like Books, Members, and Loans.
8. Explain the ACID properties of transactions and provide an example of how each property can be violated.
9. Describe two methods of concurrency control and discuss their advantages and disadvantages.
10. Write a stored procedure that takes an employee ID as input

    And returns the employee's details.

Course Code	**LMCA: 208**
Course Title	**.Net Framework and Programming in asp.net Lab**
Credits	2
Course Objective(s)	Upon completing the .NET Framework and Programming in ASP.NET Lab course, students will demonstrate proficiency in developing and deploying dynamic web applications using ASP.NET. They will gain hands-on experience with various data access techniques, implement effective state management and security practices, and creates RESTful and SOAP-based web services. Students will also develop skills in debugging, error handling, and deploying applications, while utilizing version control systems for collaborative development. Additionally, they will understand responsive web design principles, ensuring their applications are compatible across devices. Overall, students will be equipped with the practical skills and knowledge necessary for successful ASP.NET development.
Course Outcome (CO)	**CO1: Understanding the .NET Framework**: Gain a comprehensive understanding of the .NET Framework architecture, including its components, libraries, and runtime environment. **CO2**: **ASP.NET Fundamentals:** Learn the fundamentals of ASP.NET, including the differences between Web Forms, MVC, and Web API, and understand when to use each approach. **CO3: Web Application Development:** Develop skills in creating dynamic web applications using ASP.NET, focusing on best practices for coding, design, and user experience. **CO4**: **Data Access Techniques**: Understand various data access techniques in ASP.NET, including ADO.NET, Entity Framework, and LINQ, and learn how to connect to databases and perform CRUD operations **CO5**: **Debugging and Error Handling:** Develop skills in debugging ASP.NET applications and implementing effective error handling strategies to improve application reliability.

List of Lab Programs:

1. What is the .NET Framework, and what are its main components?
2. Explain the difference between ASP.NET Web Forms and ASP.NET MVC. When would you choose one over the other?
3. What is the role of the Global .asax file in an ASP.NET application?
4. How do you connect to a SQL Server database using ADO.NET? Provide a code snippet.
5. What is Entity Framework, and how does it simplify data access in ASP.NET applications?
6. Explain the purpose of LINQ and provide an example of a LINQ query to retrieve data from a list.
7. Describe the different state management techniques available in ASP.NET. When would you use each?
8. How does session state differ from view state in ASP.NET? Provide examples of when to use each.
9. What are the different authentication methods available in ASP.NET? Explain how Forms Authentication works.
10. How can you protect sensitive data in an ASP.NET application? Discuss encryption and secure connections.
11. What is the difference between RESTful and SOAP web services? Provide an example of when to use each.
12. How do you create a simple RESTful API using ASP.NET Core? Outline the steps involved.
13. What tools and techniques can you use to debug an ASP.NET application?

Course Code	**LMCA: 209**
Course Title	**Advance Java-II Lab**
Credits	2
Course Objective(s)	In Advanced Java Lab II, students will master advanced Java concepts, including multithreading and design patterns, and gain proficiency in frameworks like Spring and Hibernate. They will implement RESTful and SOAP web services, manage databases with JDBC and ORM, and develop enterprise applications using Java EE
Course Outcome (CO)	**CO1: Deepen Understanding of Java Frameworks:** Gain proficiency in popular Java frameworks such as Spring, Hibernate, and Java Server Faces (JSF) to build robust and scalable applications. **CO2**: **Master Enterprise Java Development:** Learn to develop enterprise-level applications using Java EE (Jakarta EE), including servlets, JSP, and EJB. **CO3: Implement Web Services:** Understand and implement RESTful and SOAP web services, enabling communication between different applications over the web. **CO4: Database Connectivity:** Enhance skills in database connectivity using JDBC and ORM frameworks like Hibernate, focusing on data persistence and transaction management. **CO5: Design Patterns:** Learn and apply common design patterns in Java to solve recurring design problems and improve code maintainability and scalability.

List of Lab Programs:

1. Explain the difference between Thread and Runnable.
2. How do you implement synchronization in Java?
3. What are the benefits of using the Executor Service framework?
4. Compare and contrast Array List and Linked List.
5. What is the difference between Hash Map and Tree Map?
6. How do you use streams to filter and collect data from a list?
7. Explain the concept of functional interfaces in Java.
8. Describe the steps to connect to a database using JDBC.
9. How do you handle SQL exceptions in Java?
10. What is the purpose of JPA, and how does it differ from Hibernate?
11. Explain the concept of entity relationships in JPA.
12. What are the core features of the Spring Framework?
13. How do you configure a spring application using annotations?
14. What is the difference between RESTful and SOAP web services?
15. How do you create a simple RESTful service using Spring Boot?

Course Code	**LMCA: 210**
Course Title	**Data Structure Using C++ Lab**
Credits	2
Course Objective(s)	The Data Structures using C++ lab course is designed to provide students with essential theoretical knowledge and practical skills in implementing various data structures, including arrays, linked lists, stacks, queues, trees, and graphs. By the end of the course, students will be proficient in designing and analyzing algorithms for manipulating these structures, utilizing the C++ Standard Template Library (STL) for efficient coding. They will enhance their problem-solving abilities through hands-on exercises and collaborative projects, while also developing debugging and testing skills to ensure code reliability.
Course Outcome (CO)	**CO1** :**Understanding Fundamental Concepts:** Gain a solid understanding of basic data structures, including arrays, linked lists, stacks, queues, trees, and graphs. **CO2**: **Implementation Skills**: Develop the ability to implement various data structures in C++, focusing on both static and dynamic memory allocation. **CO3: Algorithm Development:** Learn to design and implement algorithms for manipulating data structures, including searching, sorting, and traversing. **CO4**: **Problem-Solving:** Enhance problem-solving skills by applying data structures to solve real-world problems and coding challenges. **CO5: Hands-On Experience:** Provide hands-on experience through practical lab sessions, where students will implement and test various data structures and algorithms.

List of Lab Programs:

1. Write a C++ program to implement a dynamic array and demonstrate its basic operations (insertion, deletion, and traversal).
2. Implement a singly linked list in C++ and provide functions for insertion, deletion, and searching for a specific element.
3. Create a stack using an array and implement push, pop, and peek operations. Also, write a function to check if the stack is empty.
4. Implement a circular queue using a linked list in C++. Include functions for enqueue, dequeue, and displaying the queue elements.
5. Write a C++ program to create a binary tree and implement functions for in-order, pre-order, and post-order traversals.
6. Implement a binary search tree in C++ and provide functions for insertion, deletion, and searching for a node.
7. Write a C++ program to represent a graph using an adjacency list and implement depth-first search (DFS) and breadth-first search (BFS) algorithms.
8. Implement and compare different sorting algorithms (e.g., bubble sort, selection sort, and quicksort) in C++. Analyze their time complexities.
9. Create a hash table in C++ using separate chaining for collision resolution. Implement functions for insertion, deletion, and searching.
10. Implement a priority queue using a binary heap in C++. Include functions for insertion and deletion of the highest priority element.
11. Write a C++ program to implement Dijkstra's algorithm for finding the shortest path in a weighted graph.
12. Solve a problem using dynamic programming techniques, such as the Fibonacci sequence or the knapsack problem, and implement it in C++.
13. Discuss the importance of memory management in data structures and demonstrate how to manage memory dynamically in C++.
14. Analyze the time and space complexity of the data structures and algorithms implemented in the lab.

PAHER University, Udaipur
Faculty of Computer Science
LMCA(Master of Computer Applications)- 2Yr Course
SEMESTER-III

S. No.	Code	Subject Name	Period			Evaluation Scheme			
			L	T	P	ISE	ESE	Total	Credit
1	LMCA:301	Artificial Intelligence	3	1	0	30	70	100	4
2	LMCA:302	Data Warehousing & Data Mining	3	1	0	30	70	100	4
3	LMCA:303	Php	3	1	0	30	70	100	4
4	LMCA:304	Machine Learning	3	1	0	30	70	100	4
5	LMCA:305	Software Project Management	3	1	0	30	70	100	4
6	LMCA:306	Android Applications	3	1	0	30	70	100	4
7	LMCA:307	Php Lab	0	0	4	30	70	100	2
8	LMCA:308	Machine Learning Lab	0	0	4	30	70	100	2
9	LMCA:309	Android Applications Lab	0	0	4	30	70	100	2
10	LMCA:310	Software Project Management Lab	0	0	4	30	70	100	2
			7	5	12			1000	32

Course Code	**LMCA: 301**
Course Title	**Artificial Intelligence**
Credits	4
Course Objective(s)	The objective of an Artificial Intelligence (AI) course is to provide students with a solid foundation in the principles and techniques of AI, enabling them to design, develop, and implement intelligent systems. The course aims to enhance problem-solving and logical reasoning skills through the application of machine learning, natural language processing, computer vision, and robotics. It emphasizes hands-on experience with programming languages and frameworks like Python, Tensor Flow, and PyTorch, allowing students to build and evaluate AI models for real-world applications. Additionally, the course fosters an understanding of the ethical considerations and societal impacts of AI, preparing students to navigate challenges related to bias and responsible deployment. By the end of the course, students will be equipped with the knowledge and tools needed to innovate and contribute to advancements in the rapidly evolving field of artificial intelligence. .
Course Outcome (CO)	**CO1**: **Fundamental Understanding of AI:** Provide a comprehensive understanding of the fundamental concepts and techniques in artificial intelligence, including machine learning, natural language processing, computer vision, and robotics. **CO2**: **Problem-Solving Techniques**: Teach search algorithms (e.g., breadth-first, depth-first, and heuristic-based search).Discuss problem-solving paradigms such as constraint satisfaction problems and game theory. Explore optimization techniques and their application in AI. **CO3**: **Knowledge Representation and Reasoning**: Develop an understanding of knowledge representation techniques like semantic networks, ontologies, and logic-based systems. Introduce reasoning systems such as propositional and predicate logic, rule-based inference, and fuzzy logic. **CO4: Machine Learning Concepts**: Introduce supervised, unsupervised, and reinforcement learning. Familiarize students with algorithms like decision trees, support vector machines, neural networks, and clustering methods. Highlight the importance of data pre-processing and feature engineering. **CO**5: **Natural Language Processing (NLP**): Provide an overview of NLP, text processing, and language models. Discuss techniques for sentiment analysis, machine translation, and chat bots.

LMCA: 301 Artificial Intelligence

Unit -I

Introduction to AI: What is AI? , Turing Test, Typical AI problems, Practical Impact of AI, Approaches to AI, Limits of AI Today, areas of application, Aspects of AI.

Introduction to Agent: Introduction to Agents (Agent Performance, Examples of Agents, Agent Faculties, Intelligent Agents), Agent Environment, Agent architectures

Unit-II

Problem Solving using Search (Single agent search)
State space search, Search Problem, Examples (Illustration of a search process), problems with solutions of state space search (Example problem: Pegs and Disks problem, 8 queens problem, 8 puzzle, tic-tac-toe)

Search: The basic search algorithm, Search algorithm: Key issues, Evaluating Search strategies, **Blind Search:**BFS, DFS , Bi-directional search , Informed Search, Best First Search , A* search, hill climbing

Unit - III

AI Programming Language:
PROLOG: Introduction, Representation of facts, Basics of Programming, arithmetic and relational Expression, list manipulations.

Unit-IV

Natural Language Processing (NLP): Introduction, Applications of Nat. Lang. Processing, Natural language understanding,CFG's place in nlp,syntacticprocessing,definition, ingredients of cfg, illustration
Problems, Semantic Analysis.

Unit-V

Expert Systems: Introduction to expert system, Human experts v/s Expert system, Characteristics of expert system, Expert system Architecture, knowledge acquisition, case studies: MYCIN.

Recommended Books:
1. Elaine Rich and Kelvin Knight: Artificial Intelligence, Tata McGraw Hill.
2. D.W.Patterson: Introduction to Artificial Intelligence and Expert Systems, Prentice Hall of India.
3. Clocksin, W.F and Mellish, C.S: Programming in PROLOG, Narosha Publishing

Course Code	**LMCA: 302**
Course Title	**Data warehousing & Data Mining**
Credits	4
Course Objective(s)	The **Data Warehousing and Data Mining** course provides students with the knowledge to design data warehouses, perform ETL processes, and analyze data using OLAP techniques. It covers key data mining concepts like classification, clustering, and association rules, with hands-on experience using tools such as Weka and Python libraries. Students learn to apply these techniques to real-world problems while addressing data quality, ethics, and privacy. By the end of the course, they are equipped to manage and analyze large datasets effectively for informed decision-making. .
Course Outcome (CO)	**CO1: Understanding Data Warehousing Architecture:** Learn the components, architecture, and functionalities of a data warehouse. **CO2: Mastering ETL Processes:** Understand the Extract, Transform, Load (ETL) process for integrating and preparing data from multiple sources. **CO3: Schema Design Techniques:** Gain knowledge of schema models, such as star, snowflake, and fact constellation, for effective data organization. **CO4: Multidimensional Data Analysis:** Perform OLAP operations, including roll-up, drill-down, slicing, and dicing, for comprehensive data analysis. **CO5: Fundamentals of Data Mining:** Explore core concepts like classification, clustering, association rule mining, and regression. **C06**: **Implementation of Data Mining Algorithms:** Apply algorithms such as decision trees, k-means clustering, and apriori for practical problem-solving.

LMCA: 302 Data warehousing & Data Mining

Unit-I

Basic Concept of Data warehousing:-

Introduction, meaning and characteristics of data warehousing, online transaction processing(OLTP),Data warehousing model, data warehousing architecture & principles of data warehouseing and data mining.

Unit -II

Data Mining:-

What is data mining (DM)? definition and description, relationship and pattern, KDD vs data mining, DBMS vs data mining, element uses of data mining, measure data mining effectiveness:accuracy, speed, cost data information and knowledge, data mining vs machine learning, data mining model.

Unit -III

Managing and implementing a data ware housing project:
Project management process, scope statement, work breakdown structure and integration, Initiating data warehousing project estimation.

Unit -IV

Techniques of data mining:

Various technique of data mining nearest neighbour and cluster technique, decision tree, discovery of association rule ,neural network, genetic algorithm.

Unit -V

OLAP: Need for OLAP, OLAP vs. OLTP multidimensional, data model multidimensional vs. multirelational OLAP characteristics of OLAP, feature of OLAP, OLAP operation categorization of OLAP tool.

Main Text Books:

Jiawei Han &michelineKamber : “Data mining concepts and Techniques” 1st edition, first Indian reprint 2001, Harcourt India Private Limited, Isbn 1-55860489-8.

Reference Books:

1. Margaret dunham : “Data mining: Introductory and Advanced topics”, Ist edition 2003, Prentice hall (pearsonpublications) ISBMO-13-088892-3.

2. Arun k. Pujari : “Data mining Techniques”. Universities press.

Course Code	**LMCA: 303**
Course Title	**PHP**
Credits	4
Course Objective(s)	**CO1: Understanding PHP Basics**: Learn the syntax, variables, data types, operators, and control structures in PHP. . **CO2: Working with Databases**: Gain skills in connecting PHP to databases (e.g., MySQL), performing CRUD operations, and handling data. **CO3: Developing Dynamic Websites**: Learn how to create dynamic, interactive websites using PHP with HTML, CSS, and JavaScript integration. **CO4: Handling Forms and User Input**: Understand how to process and validate user input through web forms. . **CO5**:**Session Management and Cookies**: Learn how to manage user sessions and use cookies for maintaining user state. . **CO6: Error Handling and Debugging**: Develop techniques for managing errors, debugging code, and ensuring smooth application functionality. **CO7: PHP Frameworks and Tools**: Introduce popular PHP frameworks (e.g., Laravel, CodeIgniter) for developing robust web applications.
Course Outcome (CO)	Upon completing the PHP course, students will have a strong foundation in PHP programming, enabling them to develop dynamic, interactive websites. They will be proficient in connecting PHP with databases, handling user input, and managing sessions and cookies. Students will also be equipped with skills to implement security best practices and troubleshoot common issues. Additionally, they will gain experience with PHP frameworks, allowing them to build robust, scalable web applications. Overall, students will be well-prepared to create secure and efficient web solutions using PHP. .

LMCA: 303 PHP

Unit -I

Overview of PHP

1.1 Static versus Dynamic Websites
1.2 Dynamic Contents from Databases
1.3 Developing Dynamic Internet Applications
1.4 Client-Side Scripting versus Server-Side Scripting
1.5 Advantages and Capabilities of PHP
1.6 PHP versus ASP

Unit -II

Basic Scripting, Loop and Conditional Constructs

1.7 PHP Scripting Fundamentals
1.8 Primitive Data Types
1.9 Defining Constants and Variables
1.10 Loop Constructs: while, Do-while, For, Exit & Break
1.11 Conditional Constructs: If, Else, and ElseIf, Switch/Case Statement
1.12 PHP Operators: Logical, Relational, Bitwise, Ternary Operator(?)

Unit-III

Arrays in PHP

3.1 Usage of Arrays in PHP
3.2 Initializing Arrays
3.3 Adding and Removing Items from Arrays
3.4 One-dimensional and Multidimensional arrays
3.5 Array Functions

Unit-IV

Working with Forms

4.1 Form Elements
4.2 Adding Elements To a Form
4.3 Uploading Files to the Web Server Using PHP
4.4 Form Validation

Unit-V

Databases Connection

4.5 Configuring PHP for Database Support
4.6 Connection with MySql
4.7 PHP's Database API's
4.8 Database Drivers

Recommended Books:

1. Straight To the Point: PHP, Dinesh Maidasani, Laxmi Publications (Firewall)
2. Straight To the Point: MySQL, Dinesh Maidasani, Laxmi Publications (Firewall)
3. How to Do Everything with PHP & MySQL, VikramVaswani, McGraw Hills

Course Code	**LMCA: 303**
Course Title	**PHP**
Credits	4
Course Objective(s)	Upon completing the PHP course, students will be proficient in PHP programming, capable of creating dynamic web applications that integrate with HTML, CSS, and JavaScript. They will have hands-on experience with MySQL for database management and will understand security best practices to protect their applications. Familiarity with popular PHP frameworks will enhance their development skills, while collaborative projects will prepare them for real-world scenarios, equipping them for successful careers in web development.
Course Outcome (CO)	**CO1: Understand PHP Basics:** Grasp fundamental concepts of PHP, including syntax, data types, variables, and control structures. **CO2: Develop Dynamic Web Applications:** Create dynamic and interactive web applications using PHP in conjunction with HTML, CSS, and JavaScript. **CO3: Work with Databases:** Utilize PHP to connect to and manipulate databases, particularly MySQL, enabling data storage and retrieval for web applications. **CO4: Implement Security Best Practices:** Learn to secure PHP applications by implementing best practices for data validation, user authentication, and protection against common vulnerabilities. **CO5**: **Debug and Troubleshoot:** Develop skills to debug and troubleshoot PHP code effectively, ensuring robust and error-free applications.

LMCA: 303 PHP

Unit -1 Overview of PHP

1.13 Static versus Dynamic Websites
1.14 Dynamic Contents from Databases
1.15 Developing Dynamic Internet Applications
1.16 Client-Side Scripting versus Server-Side Scripting
1.17 Advantages and Capabilities of PHP
1.18 PHP versus ASP

Unit -2 Basic Scripting, Loop and Conditional Constructs

1.19 PHP Scripting Fundamentals
1.20 Primitive Data Types
1.21 Defining Constants and Variables
1.22 Loop Constructs: while, Do-while, For, Exit & Break
1.23 Conditional Constructs: If, Else, and ElseIf, Switch/Case Statement
1.24 PHP Operators: Logical, Relational, Bitwise, Ternary Operator(?)

Unit-3 Arrays in PHP

3.1 Usage of Arrays in PHP
3.2 Initializing Arrays
3.3 Adding and Removing Items from Arrays
3.4 One-dimentional and Multidimentional arrays
3.6 Array Functions

Unit-4 Working with Forms

4.9 Form Elements
4.10 Adding Elements To a Form
4.11 Uploading Files to the Web Server Using PHP
4.12 Form Validation

Unit-5Databases Connection

4.13 Configuring PHP for Database Support
4.14 Connection with MySql
4.15 PHP's Database API's
4.16 Database Drivers

Recommended Books:

4. Straight To the Point: PHP, Dinesh Maidasani, Laxmi Publications (Firewall)
5. Straight To the Point: MySQL, Dinesh Maidasani, Laxmi Publications (Firewall)
6. How to Do Everything with PHP & MySQL, VikramVaswani, McGraw Hills

Course Code	**LMCA: 304**
Course Title	**Machine Learning**
Credits	4
Course Objective(s)	Upon completing the Machine Learning course, students will have a strong understanding of machine learning algorithms and techniques. They will be able to pre-process data, build and evaluate models for tasks like classification, regression, and clustering, and optimize them for better performance. Students will also gain hands-on experience with deep learning frameworks like Tensor Flow and PyTorch. With knowledge of model evaluation and ethical considerations, they will be prepared to apply machine learning to real-world problems and make informed decisions in model development and deployment.
Course Outcome (CO)	**CO1: Understanding Machine Learning Fundamentals**: Learn the basics of machine learning, including types of learning (supervised, unsupervised, reinforcement learning) and key concepts. **CO2: Exploring Algorithms**: Study popular machine learning algorithms, such as linear regression, decision trees, k-nearest neighbours, and neural networks. **CO3: Data Pre-processing and Feature Engineering**: Gain skills in preparing and transforming data for machine learning models, including cleaning and feature selection. . **CO4**: **Model Evaluation and Optimization**: Learn how to evaluate model performance, tune hyper parameters, and apply techniques like cross-validation. **CO5**: **Applying Machine Learning to Real-world Problems**: Understand how to implement machine learning solutions for tasks such as classification, regression, and clustering. .

LMCA: 304 MACHINE LEARNING

Unit – I

Introduction Machine Learning Foundations – Overview – Design of a Learning System – Types of Machine Learning – Supervised Learning and Unsupervised Learning – Mathematical Foundations of Machine Learning – Applications of Machine Learning.

Unit – II

Supervised Learning - I Simple Linear Regression – Multiple Linear Regression – Polynomial Regression – Ridge Regression – Lasso Regression – Evaluating Regression Models – Model Selection – Bagging – Ensemble Methods.

Unit – III

Supervised Learning - II Classification – Logistic Regression – Decision Tree Regression and Classification – Random Forest Regression and Classification – Support Vector Machine Regression and Classification - Evaluating Classification Models.

Unit – IV

Unsupervised Learning Clustering – K-Means Clustering – Density-Based Clustering – Dimensionality Reduction – Collaborative Filtering.

Unit – V

Association Rule Learning and Reinforcement Learning Association Rule Learning – Apriori – Eclat – Reinforcement Learning – Upper Confidence Bound – Thompson Sampling – Q-Learning.

REFERENCES:

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.

2. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Third Edition, 2014.

3. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997.

4. Sebastian Raschka, Vahid Mirjilili,”Python Machine Learning and deep learning”, 2nd edition, kindle book, 2018

5. Carol Quadros,”Machine Learning with python, scikit-learn and Tensorflow”, Packet Publishing, 2018

6. Gavin Hackeling,” Machine Learning with scikit-learn”, Packet publishing, O’Reily, 2018

7. Stanford Lectures of Prof. Andrew Ng on Machine Learning

Course Code	**LMCA: 305**
Course Title	**Software Project Management**
Credits	4
Course Objective(s)	Upon completing the Software Project Management course, students will be equipped with the skills to successfully manage software projects from start to finish. They will be able to plan and schedule projects, assess and manage risks, and lead teams effectively. Students will also gain expertise in quality assurance, cost management, and resource allocation. With knowledge of both Agile and traditional methodologies, they will be able to select and implement the appropriate approach for various projects. Overall, students will be prepared to ensure that software projects are completed on time, within scope, and within budget. .
Course Outcome (CO)	**CO1: Understanding Software Project Life Cycle**: Learn the stages of a software project, from initiation to planning, execution, monitoring, and closure. . **CO2: Project Planning and Scheduling**: Gain skills in project estimation, creating schedules, and defining project scope, resources, and deliverables. . **CO3: Risk Management**: Understand how to identify, assess, and mitigate risks throughout the project lifecycle. **CO4**: **Team Management and Communication**: Learn how to lead project teams, assign tasks, manage conflicts, and ensure effective communication among stakeholders. **CO5**: **Quality Assurance and Testing**: Understand the importance of quality management, testing strategies, and ensuring software quality throughout development. **CO6: Project Monitoring and Control**: Learn how to track project progress, handle changes, and ensure projects are completed on time and within scope.

LMCA: 305 Software Project Management

Unit -I

Introduction, software life-cycle models, software requirements specification, formal requirements specification, verification and validation.

Unit -II

Software Project Management: Objectives, Resources and their estimation, LOC and FP estimation, effort estimation, COCOMO estimation model, risk analysis, software project scheduling.

Unit -III

Requirement Analysis: Requirement analysis tasks, Analysis principles. Software prototyping and specification data dictionary, Finite State Machine (FSM) models. Structured Analysis: Data and control flow diagrams, control and process specification behavioral modelling

Unit -IV

Software Design: Design fundamentals, Effective modular design: Data architectural and procedural design, design documentation.

Unit -V

Object Oriented Analysis: Object oriented Analysis Modelling, Data modelling. Object Oriented Design: OOD concepts, Class and object relationships, object modularization, Introduction to Unified Modelling Language

Text Book:

1 Software Project Management, Bob Huges,Mike Cotterell,Rajib Mall 5th Edition

2 Software Project Management, by Dutt, Chandramouli Released December 2015Publisher(s): Pearson India ISBN: None

3 Introduction to Software Project Management, By Adolfo Villafiorita

Course Code	**LMCA: 306**
Course Title	**Android Applications**
Credits	4
Course Objective(s)	Upon completing the Android Applications course, students will have the skills to design, develop, and deploy Android apps effectively. They will be proficient in creating user-friendly interfaces, managing app data with databases, and integrating web services and APIs. Students will also gain expertise in handling the Android app lifecycle, optimizing performance, and incorporating multimedia elements. Additionally, they will be prepared to test, debug, and publish Android apps on platforms like Google Play. Overall, students will be capable of developing fully functional, high-quality Android applications.
Course Outcome (CO)	**CO1**: **Understanding Android Fundamentals**: Learn the basics of the Android operating system, its architecture, and key components like Activities, Services, and Broadcast Receivers. **CO2: User Interface Design**: Gain proficiency in designing user-friendly interfaces using Android UI components, layouts, and themes. . **CO3: Working with Databases**: Learn how to integrate databases (e.g., SQLite, Firebase) to store and manage app data. **CO4**: **App Lifecycle Management**: Understand the lifecycle of Android applications, including managing app states and memory efficiently. **CO5**: **Networking and Web Services**: Develop skills in integrating Android apps with web services and APIs to fetch and display remote data. **CO6: Multimedia Integration**: Learn how to incorporate multimedia elements like images, audio, and video into Android apps.

LMCA: 305 Android Applications

Unit -I

Basics of Android: What is Android? , History of android, Version of android, Features of android Setting up development environment, Android Architecture, Android application, Anatomy of android application.

Unit -II

Working with the User Interface:- Development tools, Applications (App manifest , Resources ,Application types), Activities(Activity Life Cycle),Introduction to UI(Layouts , Adapters ,Action bar , Dialogs Notifications).

Unit -III

UI Widgets: Working with Button, Toast, Custom Toast, Toggle Button, Switch Button, Image Button, Checkbox, Alert Dialog, Rating Bar, Date Picker, Time Picker, Progress Bar .

Unit -IV

Intent & Fragment, Menu, Layout Manager and view: implicit Intent, Explicit Intent, Fragment Lifecycle ,Fragment Example ,Dynamic Fragment ,Option Menu, Context Menu ,Popup Menu ,Relative Layout, Linear Layout, Table Layout, Grid Layout, GridView, Web View, Scroll View, Search View

Unit -V

Android Service: Android Service, Android Service API, Android Service Life Cycle.
SQLite: introduction to SQLite, Database connectivity with SQLite

Recommended Books:

1. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011)

Reference Books:

1. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd (2011)

2. Mark L Murphy, “Beginning Android”, Wiley India PvtLtd(2009)
3. Sayed Y Hashimi and SatyaKomatineni, “Pro Android”, Wiley India PvtLtd(2009

Course Code	**LMCA: 307**
Course Title	**PHP Lab**
Credits	2
Course Objective(s)	Upon completing the PHP Lab course, students will be proficient in PHP programming and capable of developing dynamic web applications. They will learn to integrate PHP with HTML, CSS, and JavaScript, as well as connect to databases like MySQL for data management. The course will cover form handling, session and cookie management, and object-oriented programming principles. Students will also gain exposure to popular PHP frameworks, understand security best practices, and develop debugging and testing skills. Through hands-on projects, they will demonstrate their ability to create functional web applications, preparing them for careers in web development.
Course Outcome (CO)	**CO1: Fundamental PHP Skills:** To provide students with a solid understanding of PHP syntax, data types, and control structures. **CO2: Dynamic Web Development:** To enable students to create dynamic and interactive web applications using PHP in conjunction with HTML, CSS, and JavaScript.. **CO3: Database Connectivity:** To teach students how to connect PHP applications to databases (e.g., MySQL) for data storage, retrieval, and manipulation. **CO4: Form Processing:** To equip students with the skills to create, validate, and process web forms effectively. **CO5**: **Session and Cookie Management:** To familiarize students with managing user sessions and cookies for maintaining state in web applications. **CO6: Object-Oriented Programming:** To introduce students to object-oriented programming concepts in PHP, promoting code reusability and organization. **CO6: Project Implementation:** To encourage hands-on experience through projects that allow students to apply their knowledge and build functional web applications.

List of Lab programs:

1. Write a PHP script to declare variables of different data types (string, integer, float, Boolean) and display their values.
2. Create a PHP program that uses an if-else statement to check if a number is even or odd.
3. Write a PHP script that uses a for loop to print the numbers from 1 to 10.
4. Define a function that takes two numbers as parameters and returns their sum. Call this function and display the result.
5. Create an associative array to store the names and ages of five people. Write a loop to display each name with its corresponding age.
6. Create an HTML form that collects a user's name and email. Write a PHP script to process the form and display the submitted information.
7. Extend the previous form to include validation for the email format and ensure that the name is not empty.
8. Write a PHP script to connect to a MySQL database and retrieve all records from a table named users.
9. Create a PHP script that inserts a new user into the users table using data from a form.
10. Write a PHP script that starts a session, stores a user's name in the session, and displays a welcome message.

Course Code	**LMCA: 308**
Course Title	**Machine Learning Lab**
Credits	2
Course Objective(s)	Upon completing the Machine Learning Lab course, students will have gained practical experience in implementing and applying machine learning algorithms to real-world problems. They will be proficient in data pre-processing, model evaluation, and optimization techniques. Students will be able to use machine learning libraries like Scikit-learn, Tensor Flow, and Keras to build and fine-tune models. Additionally, they will have hands-on experience with deep learning frameworks and data visualization tools, enabling them to analyze and interpret model results effectively. Overall, students will be equipped to tackle machine learning challenges in real-world applications.
Course Outcome (CO)	**CO1: Implementing Machine Learning Algorithms**: Gain practical experience in coding and applying popular machine learning algorithms such as linear regression, decision trees, k-means clustering, and neural networks. **CO2: Data Pre-processing and Feature Engineering**: Learn how to clean, pre-process, and transform real-world datasets for machine learning models. **CO3: Model Evaluation and Optimization**: Practice evaluating the performance of models using various metrics and techniques like cross-validation, hyper parameter tuning, and model selection. . **CO4**: Working **with Machine Learning Libraries**: Use industry-standard libraries and frameworks such as Sickest-learn, Tensor Flow, and Keras to build machine learning solutions. . **CO5**: **Deep Learning Fundamentals**: Gain exposure to deep learning techniques and frameworks like TensorFlow and PyTorch for building neural networks.

List of Programs:

1. Load and explore the dataset.
2. Preprocess the data by handling missing values and normalizing the features if necessary.
   Split the data into training and testing sets.
1. Implement and train a classification model (e.g., K-Nearest Neighbors, Support Vector Machine, or Decision Tree).
2. Evaluate the model's performance using accuracy, precision, recall, and F1-score.
3. Visualize the decision boundaries for the model.
4. Apply cross-validation and hyper parameter tuning to improve model performance.

Course Code	**LMCA: 309**
Course Title	**Android Applications Lab**
Credits	2
Course Objective(s)	The Android Applications Lab course equips students with the skills to develop Android applications. Students will learn to set up the development environment, design user interfaces, manage activities, implement data storage, and make network requests. The course also covers multimedia integration, location services, background services, testing, debugging, and publishing apps on the Google Play Store, emphasizing best practices and collaboration in software development.
Course Outcome (CO)	**CO1: Understand Android Development Fundamentals:** Learn the architecture and components of the Android platform, including Activities, Services, Broadcast Receivers, and Content Providers. Gain familiarity with Android Studio as the primary development environment. **CO2: Learn User Interface Design:** Develop responsive and user-friendly interfaces using XML layouts and Java/Kotlin code. Implement advanced UI features such as menus, dialogs, navigation drawers, and fragments. **CO3: Explore Android Framework and APIs :** Work with Android libraries for data storage, network communication, and media handling. Integrate APIs for location-based services, camera usage, and sensor data. **CO4: Develop Application Logic and Functionality:** Build functional apps with robust back-end logic, including database integration using SQLite or Room. Handle lifecycle events, background tasks, and multithreading. **CO5**: **Debugging and Testing:** Learn debugging techniques to resolve issues in Android applications. Test apps using tools like JUnit and Android Emulator. **CO6: Collaborative and Project-Based Learning:** Work on team-based projects to develop real-world Android applications. Build a portfolio showcasing hands-on experience with Android development.

List of Lab programs:

1. How to set up Android Studio and create a "Hello World" application?
2. How to demonstrate the lifecycle of an Activity using log messages?
3. How to design a login screen with username, password, and login/reset buttons?
4. How to implement explicit and implicit intents in an Android app?
5. How to handle button click events in an app?
6. How to create a registration form using different layouts and Scroll View?
7. How to implement navigation between fragments in an Android app?
8. How to create a dynamic list using Recycler View?
9. How to add an options menu, context menu, and custom dialogs in an app?
10. How to store and retrieve data using Shared Preferences, SQLite, or Room database?
11. How to play audio or video files in an app using Media Player?
12. How to fetch and display data from a RESTful API using Retrofit or Volley?
13. How to use accelerometer sensors to implement a shake feature in an app?
14. How to display the user's current location using Google Maps API?
15. How to create and manage notifications using AlarmManager or Work Manager?
16. How to create a To-Do list app with CRUD operations?
17. How to develop a weather app that fetches real-time weather data?
18. How to build a fitness tracker using sensors to track steps and calories?
19. How to develop a basic e-commerce app with product listing and a shopping cart?
20. How to design a social media app with user registration and post display?

Course Code	**LMCA: 310**
Course Title	**Software Project Management Lab**
Credits	2
Course Objective(s)	The software project management lab enable students to plan , schedule, manage software projects effectively. They gain skills in risk analysis, cost estimation, teamwork using tools like JIRA or MS projects, preparing them for real world project management challenges.
Course Outcome (CO)	**CO1: Project Planning and Scheduling**: Gain hands-on experience in defining project scope, creating schedules, and estimating resources using project management tools like Microsoft Project or Jira. **CO2: Risk Management**: Learn how to identify, assess, and mitigate risks in real-time project scenarios. . **CO3: Team Collaboration**: Develop skills in team management, communication, and collaboration using project management software and techniques. **CO4: Quality Management**: Apply techniques for ensuring the quality of software projects through testing strategies, reviews, and process improvements. . **CO5**: **Tracking and Reporting**: Learn how to monitor project progress, track milestones, and generate reports on time, cost, and quality. **CO6: Resource Management**: Understand resource allocation, budgeting, and cost management, ensuring efficient use of time, personnel, and financial resources.

List of Lab Programs:

1. Create Gantt chart for given software project.
2. Develop a work breakdown structure (WBS) for project Scenario.
3. Estimate the cost of a software project using function point analysis.
4. Identify and document risks for a software project and propose mitigation strategies.
5. Allocate resources for project and optimize resource utilization.

Pacific University, Udaipur
Faculty of Computer Application

LMCA (Master of Computer Applications)- 2Yr Course

SEMESTER-IV

S. No.	Code	Subject Name	Period			Evaluation Scheme			
			L	T	P	ISE	ESE	Total	credit
1	LMCA: 401	**Major Project**				200	300	500	24

Course Code	**LMCA: 401**
Course Title	**Major Project**
Credits	2
Course Objective(s)	The **Major Project** course enables students to apply theoretical knowledge to practical problems, develop solutions, and enhance their technical and teamwork skills. By completing the project, students gain experience in project design, implementation, documentation, and effective communication, preparing them for real-world challenges.
Course Outcome (CO)	**CO1: Application of Knowledge**: To apply theoretical concepts learned during the course to solve real-world problems through the development of a functional project. **CO2: Research Skills Development**: To encourage students to conduct research, explore relevant literature, and stay updated with the latest technologies and methodologies related to their project topic. **CO3: Project Design and Implementation**: To enable students to design, develop, and implement a project based on their chosen topic, showcasing their ability to integrate various components of their field of study. **CO4: Problem-Solving and Critical Thinking**: To develop problem-solving skills by identifying issues, proposing solutions, and analyzing outcomes during the course of the project. **CO5: Technical Skills Enhancement**: To enhance students' technical skills by working on software, hardware, or systems relevant to their area of study, and applying hands-on techniques to build, test, and refine prototypes. **CO6: Teamwork and Collaboration**: To promote teamwork by encouraging collaborative work, where students contribute to various aspects of the project and learn to manage responsibilities and deadlines effectively. **CO7: Report Writing and Documentation**: To train students in preparing detailed project reports and documentation, including the ability to present their findings, methodologies, and outcomes in a structured and professional manner. **CO8: Presentation and Communication Skills**: To improve students' ability to present their projects clearly and confidently, both in written and oral formats, to peers, faculty, or external stakeholders. **CO9: Time and Resource Management**: To help students manage time and resources effectively throughout the project's lifecycle, ensuring timely completion within the constraints of the course.