Syllabus

Master of Computer Science (Msc.IT)

2024-25

"Pacific Hills", Pratap Nagar Extension, Airport Road, Debari, Udaipur (Raj.) | Ph. : 0294-2494520 Fax : 0294-2492641
Email : fca@pacific-university.ac.in | www.pacific-university.ac.in

Program Outcomes for Msc.IT

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) Apply the theoretical and practical knowledge of computer science in formulating, modelling and developing solutions to the real world problems.

2) Analyze Design and implement the application software systems that meet the automation requirement of society and industry

3) Ability to apply knowledge of layered network Models, their protocols and technologies in building network and Internet based applications.

4) Design, test, develop and maintain desktop, web, mobile and cross platform software applications using modern tools and technologies.

Program Education Outcome (PEO)

PSO1: An ability to enhance the application of knowledge of theory subjects in diverse fields.

PSO2: Develop language proficiency to handle corporate communication demands.

PSO3: Preparing students in various disciplines of technologies such as computer applications, computer networking, software engineering, JAVA, database concepts and programming.

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.

Pacific University, Udaipur
Faculty of Computer Application
Msc.IT(Master of Computer Science)- 2Yr Course
SEMESTER-I

S. No.	Code	Subject Name	Period			Evaluation Scheme		
			L	T	P	ISE	ESE	Total
1.	MCS:101	Accounting &financial management	3	1	0	30	70	100
2.	MCS:102	Information Security & S/W Standards	3	1	0	30	70	100
3.	MCS:103	Network Management	3	1	0	30	70	100
4.	MCS:104	Programming Through C++	3	1	0	30	70	100
5.	MCS:105	Web Programming	3	1	0	30	70	100
6.	MCS:151	Programming Through C++ Lab	0	0	4	30	70	100
7.	MCS:152	Web Programming Lab	0	0	4	30	70	100
8.	MCS:153	Seminar	0	0	4	30	70	100
			15	5	12			800

Course Code	**MCS:101**
Course Title	**Accounting & financial management**
Credits	4
Course Objective(s)	The Accounting and Financial management course equips student with the skills to prepare and analyze financial statement, mange budget and control cost. It fosters decision making abilities through investment evaluation and risk management techniques. Student will also learn strategic financial planning, corporate finance concept, and ethical compliance. Additional the course emphasizes the use of financial tools and technology for accurate reporting and effective decision making in a business context.
Course Outcome (CO)	**CO1**: **Understanding Financial Concepts**: Gain foundational knowledge of accounting principles, financial statement, and financial management techniques. **CO2: Financial Statement Analysis:** Learn to interpret and analyse balance sheet, income statement, and cash flow. **CO3**: **Cost Management**: Understand cost behaviour, budgeting, and method to control and reduce costs. **CO4: Decision and Planning:** Develop skills to make informed financial decisions using tools like break-even analysis and financial forecasting. **CO5**: **Budgeting and planning:** Learn techniques for budgeting, financial planning, and performance evaluation.

MCS:101 Accounting & Financial Management

UNIT – I

Definition of Accounting and its advantages & limitations, Scope of accounting, Branches of Accounting – Financial Accounting

UNIT – II

Types of Accounts and Rules for Debit and Credit. Cash and Credit Transaction, Cash discount and trade discount. Preparation of Journal, ledger.

UNIT – III

Business and Management: Business Meaning and Contents, Business as a system,

Business Environment. Management Concept and Nature, Management Process, Basic function of Management, Management Level, Role of Manager.

UNIT – IV

Organizational Behaviour: Need of Understanding human behaviour in organization, Challenges and Opportunities for OB. Management by Objective (MBO

UNIT- V

Managing Personnel: Motivation Theories & Practices, Leadership Concept theories & Style, Compensation Management.

Recommended Books:

1. B.P. Singh & T.N. Chabra, "Business Organization and Management Functions", DhanpatRai& Co. 2000.

2. P.C Tripathi& P. N. Reddy, "Principles of Management", Tata McGraw Hill Publishing Company New Delhi.

3. L.M. Prasad, "Principles and Practices of Management".

Course Code	**MCS:102**
Course Title	**Information-Security and Software Standards**
Credits	4
Course Objective(s)	Upon completing information security and software standard course, students will be able to identify and mitigate security threats, implement risk management strategies and apply cryptographic techniques to protect data . They will understand secure network protocols, access control mechanisms and compliance with security policies. Additionally, students will gain proficiency in integrating security within the software development life cycle and adhering to industry standards. The course also prepares students to handle security incidents and navigate the ethical and legal aspects of information security.
Course Outcome (CO)	**CO1**: **Fundamental of Information Security**: Understand the principles of confidently, integrity, and availability in securing information. **CO2: Threats and Risk Management:** Identify security threats, vulnerability, and implement risk management strategies. **CO3: : Network Security private network:** Understand network security protocols, firewalls, intrusion detection system, and virtual **CO4**: **Cryptography Techniques:** Learn cryptography methods and their application in securing data transmission and storage. **CO5**: **Access Control and Authentication:** Explore methods for controlling access and ensuring user authentication. **CO6**: **Software Security Standards:**Familiarize with industry standards and best practices for secure software development. **CO7**: **Security Policies and Compliance:**Develop and implement security policies in line with regulatory and compliance requirement. **CO8**: **Incident Response and Recovery:** Learn strategies for detecting, responding to, and recovering from security breaches. **CO9**: **Secure Software Development Life Cycle:** Integrated security practices into each phase of the development process. **CO10**: **Ethical and Legal Consideration:** Understand ethical issue, legal frameworks and professional responsibilities in information Security.

MCS: 102Information-Security and Software Standards

UNIT I

Information Security: Introduction, History of Information security, What is Security, Components of Information System, Balancing Information Security and Access, Approaches to Information Security Implementation, The Security Systems Development Life Cycle.

UNIT II

Intruders, Virus and Firewalls: Intruders, Intrusion detection, password management, Virus and related threats -Malware, Phishing, DoS, Hacking, Bots, Worms, Trojans and Spam, Firewall design principles, Types of firewalls.

UNIT III

System & Web Security: Cryptography: Concepts and Techniques, symmetric and asymmetric key cryptography. Web Security: Web Security Considerations, Secure socket layers and Transport Layer Security, Secure Electronic Transaction E-mail security: Pretty Good Privacy, S/MIME (Secure/ Multipurpose Internet Mail Extension).

UNIT IV

Software Standards: Introduction to Software Quality, Software Quality in Business Context, Managing Software Quality in an Organization, Software Quality Assurance(SQA) Plan – Overview, Purpose and Scope, SQA Management, Product Quality and Process Quality, Software Measurements and Metrics.

UNIT V

Software Quality Improvement Models: Basics of ISO 9001 and Software Capability Maturity Model, Comparison of SEI-CMM and ISO 9000 for Software, The Sixth Sigma Model, Managing the Software Process Improvement Project.

TEXT BOOK:

1. Michael E Whitman and Herbert J Mattord, “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003

REFERENCES:

1. Micki Krause, Harold F. Tipton, “ Handbook of Information Security Management”, Vol 1-3 CRC Press LLC, 2004.

2. Stuart McClure, Joel Scrambray, George Kurtz, “Hacking Exposed”, Tata McGraw-Hill, 2003

3. Matt Bishop, “Computer Security Art and Science”, Pearson/PHI, 2002.

Course Code	**MCS:103**
Course Title	**Network Management**
Credits	4
Course Objective(s)	The **Network Management course** equips learners with the skills to design, configure, monitor, and secure computer networks. Participants gain expertise in using network tools, troubleshooting issues, and implementing security measures such as firewalls and VPNs. The course also covers emerging technologies like SDN and IoT, ensuring adaptability to industry trends. Through hands-on projects, learners acquire practical experience, preparing them for certifications like CCNA and careers in network administration or management.
Course Outcome (CO)	**CO1: Understanding Network Fundamentals:** Gain Knowledge of network architecture, protocols and topologies. **CO2: Network Planning and Design:** Learn to design, configure and optimize networks to meet organizational needs. . **CO3: Network Monitoring and Performance:** Understand tools and techniques for monitoring network performance and ensuring reliability. **CO4: Fault Management: Develop skills to detec**t, diagnose and resolve network issue efficiently. **CO5: Security Management:** Implement network security measures, including firewalls, intrusion detection System and encryption. **CO6: Configuration and Change Management:** Learn best practices for managing network Configuration and tracking changes.

MCS: 103 Network management

Unit I

Data Communication : Basics of Data Communications, Components of Communication, Communication Model, Data Representation, Communication Modes, Network Basics & criteria, uses of computer networks, physical structure, network topology, Basics of LAN, WAN, MAN, Internet, Protocols and Standards.

Unit II

OSI Model T &CP/IP Model and Protocols: Layered Tasks, Layered Architecture, Design Issues for the Layer, OSI Model- Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer. TCP/IP Model, TCP/IP Protocols, Addressing- Logical Addressing, Physical Addressing. Connection Oriented and Connection less Services.

Unit III

Physical Layer and Media & Data Link Layer: Data and Signals, Transmission Impairment and Performance, Multiplexing and Spreading, Transmission Media, Switching, Introduction to PSTN. Introduction to Error, Types of Error, Error Detection and Correction.

Unit IV

Network Layer & Transport Layer : IPv4 Addresses, IPv6 Addresses, Comparison b/w IPv4 and IPv6, Routing, Introduction to Routing Protocols. Flow Control, Buffering, UDP, TCP. Standards-IEEE standard, Ethernet, IEEE 802.11, Bluetooth, Wireless LANs. Connecting Devices, Virtual LAN.

Unit V

Application Layer & Network Management and Security: DNS, Remote Logging, E-Mail, File Transfer, WWW, HTTP. Network Management System, SNMP, VPN, Cyber Laws.

Recommended Books:

1. A. S Tanenbaum, “Computer Networks, 3rd Edition”, PHI
2. W. Stallings, “Data and Computer Communication”, Macmillan Press
3. Comer, “Computer Networks & Internet”, PHI.
4. Comer, “Internetworking with TCP/IP”, PHI
5. Frozen, “Data Communication and Networking”, TMH

Course Code	**MCS:104**
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.

MCS: 104 Programming through C++

UNIT I

Overview of OOP: Definition and need for OOP, Differences between OOP and Procedure-Oriented Programming, Key principles of OOP: Abstraction, Classes and Objects, Encapsulation, Inheritance and Data Binding, Polymorphism, C++ Basics, Structure of a C++ program, Data types ,Declaration of variables, Expressions and Operators, Type Conversions ,Introduction to C++ Streams ,Flow Control Statements

Functions: Classification of Functions, Parameter Passing, Recursive Functions, Default Arguments, Inline Functions, Arrays and 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 functions and classes ,Static class members, Overloading ,Function Overloading. Operator Overloading**,** Unary Operators, Binary Operators.

UNIT III

Inheritance: Defining a Class Hierarchy, Different forms of inheritance, Defining base and derived classes, Access to 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, Benefits of exception handling ,Throwing an exception, The try block, Catching an exception.

UNIT V

Advanced Topics: Templates, Function Templates, Class Templates, File Handling, Stream classes hierarchy, Stream/O, File streams, Opening and closing data files, creating a data file, Reading and writing files, Error handling during file operations

Recommended Books:

1. BjarneStroustrup, “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
6. R. Kruse etal, “Data Structures and Program Design in C” Pearson Education
7. A M Tenenbaumetal, “Data Structures using C & C++”, PHI

Course Code	**MCS: 105**
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**: **CO4: 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. **CO4: Version Control and Collaboration:** Learn to use version control systems like Git for collaborative development and project management.

MCS: 105Web Programming

Unit-1

The internet: Worldwide 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-2

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-3

Dynamic HTML: object model and collections: introduction, Dynamic style, Dynamic Positioning.

Dynamic HTML: event model : introduction, event ON CLICK, event ON LOAD

Unit-4

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-5

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	**MCS: 151**
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:

Student Grade Calculator: Write a program that calculates the average grade of a class of students based on user input.

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

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

Sorting an Array: Implement a bubble sort algorithm to sort an array of integers input by the user.

File I/O - Word Count: Write a program that reads a text file and counts the number of words in it, prompting the user for the filename.

Temperature Converter: Write a program that converts temperatures between Celsius and Fahrenheit based on user input.

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

Matrix Multiplication: Write a program that multiplies two matrices input by the user and displays the result.

Prime Number Checker: Write a program that checks if a given number is prime.

Contact Management System: Create a program that allows users to add, view, and delete contacts from a simple contact list.

Course Code	**MCS: 152**
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. .

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	**MCS: 153**
Course Title	**seminar**
Credits	2
Course Objective(s)	The seminar course aims to enhance students' critical thinking, communication, and collaboration skills while engaging with complex topics. By the end of the course, students will demonstrate the ability to analyze diverse perspectives, articulate their ideas clearly, and work effectively in teams. They will develop research proficiency by utilizing credible sources and reflect on ethical considerations related to the subject matter. Through self-reflection, students will identify areas for personal growth and apply their knowledge to real-world situations, ensuring their learning extend beyond the classroom and prepare them for future challenges.
Course Outcome (CO)	**CO1**: **Critical Thinking and Analysis** Develop the ability to analyze complex issues and arguments. Encourage students to evaluate different perspectives and synthesize information from various sources. **CO2:Communication Skills** Enhance verbal and written communication skills through presentations and discussions. Foster the ability to articulate ideas clearly and persuasively in both individual and group settings. **CO3:Collaboration and Teamwork** Promote collaborative learning by engaging in group projects and discussions. Encourage respect for diverse viewpoints and the ability to work effectively with **peers.CO4:Research and Inquiry** Cultivate research skills by requiring students to gather, evaluate, and utilize information from credible sources. Encourage independent inquiry and critical questioning of established knowledge. **CO5:Personal and Social Responsibility** Foster a sense of ethical responsibility and civic engagement through discussions on relevant social issues. Encourage reflection on personal values and their impact on society.

Pacific University, Udaipur
Msc.IT (Master of Computer Science)- 2Yr Course
Faculty of Computer Application
SEMESTER-II

S.No.	Code	Subject Name	Period			Evaluation Scheme		
			L	T	P	ISE	ESE	Total
1.	MCS: 201	Advanced Database Management System-II	3	1	0	30	70	100
2.	MCS: 202	.Net Framework and Programming in asp.net	3	1	0	30	70	100
3.	MCS: 203	Advance Java-II	3	1	0	30	70	100
4.	MCS: 204	Distributed System	3	1	0	30	70	100
5.	MCS: 205	Data Structure Using C++	3	1	0	30	70	100
6.	MCS: 251	Advanced Database Management System-II Lab	0	0	4	30	70	100
7.	MCS: 252	.Net Framework and Programming in asp.net Lab	0	0	4	30	70	100
8.	MCS: 253	Advance Java-II Lab	0	0	4	30	70	100
9.	MCS: 254	Data Structure Using C++ Lab	0	0	4	30	70	100
			15	5	16			900

Course Code	**MCS: 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: Explore Advanced Topics:** Investigate critical topics such as database security, data integrity, and emerging trends in database. Students to analyse 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**.**

MCS: 201Advanced 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, Reference 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, inter query parallelism, intra query parallelism, intra operation parallelism, intra operation 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

Recommended Books:

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. Ramakrishna, Gehrke, "Database Management System", McGraw Hill.

Course Code	**MCS: 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**.**

MCS: 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.

Recommended Books:

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	**MCS: 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 micro services 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.

MCS: 203Advanced 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, connection pooling, 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, JSP Declarations, JSP Directives, JSP Actions, 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	**MCS: 204**
Course Title	**DISTRIBUTED SYSTEMS**
Credits	4
Course Objective(s)	Upon completing the Distributed Systems course, students will be equipped to analyze and implement various distributed system architectures, design effective communication protocols, and ensure data consistency and fault tolerance. They will optimize application performance, address security challenges, and gain hands-on experience in developing real-world distributed applications using modern tools. Additionally, through collaborative projects, students will enhance their teamwork and communication skills, preparing them to tackle complex challenges in distributed computing and contribute effectively to the development of sophisticated systems in diverse industries.. 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**: **Understand Fundamental Concepts:** Grasp the key concepts and challenges associated with distributed systems, including communication, synchronization, fault tolerance, and scalability.. **CO2: Explore Architectures:** Analyse various distributed system architectures, such as client-server, peer-to-peer, and micro services, and understand their use cases and trade-offs. **CO3: Implement Communication Protocols:** Learn about different communication models and protocols, including Remote Procedure Calls (RPC), message passing, and data streaming, and implement them in practical scenarios. **CO4: Manage Data Consistency:** Understand data consistency models, including eventual consistency and strong consistency, and explore techniques for managing distributed data storage and replication. **CO5**: **Ensure Fault Tolerance:** Study methods for achieving fault tolerance and reliability in distributed systems, including redundancy, consensus algorithms, and recovery techniques.

MCS: 204DISTRIBUTED SYSTEMS

UNIT I

Introduction to Distributed System, Examples of Distributed Systems, Characteristics, Trends in Distributed Systems, Focus on resource sharing, Challenges in Distributed system. Distributed Systems Architecture: software layers, Architectural Models-Client-server, Peer processes

UNIT II

Communication In Distributed System- System Model, Inter process Communication, API for internet protocols ,Types of network, network principles, Internet protocols, Network case studies - Ethernet, wireless LAN and ATM, Client-server communication, Group Communication.

UNIT-III

Object Interaction: RMI, RPC, Distributed Application Programming: Distributed object models, Operating System Support: The Operating System Layer, Protection, Processes and Threads. Distributed File Systems: File Service Architecture, Sun network File System, the Andrew File System.

UNIT-IV

Transactions and Concurrency Control: Transaction, Nested transactions, Locks, Optimistic Concurrency control, Time stamp ordering and Comparisons of methods for concurrency controls. Distributed Transactions - Flat and nested distributed transactions, Atomic commit protocols, and Concurrency control in distributed transactions, distributed deadlocks, Transaction Recovery.

UNIT-V

Distributed Multimedia System: Characteristics of multimedia data, Quality of Service management, Resource management, Stream Adaptation. Distributed Shared Memory Design and implementation issues.

Recommended Books:

1. George Coulouris, Jean Dollimore and Time Kindberg, Distributed Systems: Concepts and Design, Third Edition 2002, Pearson Education Asia.

2. Andrew S. Tanenbaum& Maarten van Steen, Distributed Systems: Principles and Paradigms, 2002 Prentice-Hall.

3. R. Chow and T. Johnson: “Distributed Operating Systems & Algorithms”, 1997, Add isonWesley.

4. D. L. Galli: “Distributed Operating Systems”, 2000, Prentice Hall.

5. C. Leopold: “Parallel and Distributed Computing”, 2001, John Wiley & Sons.

6. M. Raynal, J. Howlett, trans. : “Distributed Algorithms and Protocols”, 1998, J. Howlett, trans., Wiley & Sons.

7. P. K. Sinha: “Distributed Operating Systems: Concept and Design”, 1997, IEEE Press.

Course Code	**MCS: 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 analyse 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.

MCS: 205 Data Structures Using C++

UNIT-I

Arrays, records, stack, operation on stack, implementation of stacks an array, 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, inserting into an 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

Graph Structure: Graph representation-Adjacent matrix, adjacency list. Orthogonal representation of graph. Graph traversals bfs and dfs. Shortest path, 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, Fundamentals of Data Structure, Galgotia Publication Private Limited., New Delhi.
2. R.S. Saleria, Data Structure and algorithm using C.
3. R.B. Patel, Expert data structure with C.

Course Code	**MCS: 251**
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. 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, sub queries, 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 modelling, 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	**MCS: 251**
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 create 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	**MCS: 253**
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.
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: Concurrency and Multithreading:** Explore advanced concepts of concurrency and multithreading in Java, including thread management, synchronization, and concurrent collections. **CO6: 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	**MCS: 254**
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. **CO6**: **Debugging and Testing:** Develop skills in debugging and testing C++ programs, ensuring the correctness and efficiency of data structure implementations.

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.
15. Discuss a real-world application of a specific data structure and implement a simple version of it in C++.

Pacific University, Udaipur
Faculty of Computer Application
Msc.IT (Master of Computer Science)- 2Yr Course
SEMESTER-III

S. No.	Code	Subject Name	Period			Evaluation Scheme		
			L	T	P	ISE	ESE	Total
1.	MCS: 301	Artificial Intelligence	3	1	0	30	70	100
2.	MCS: 302	Data Warehousing & Data Mining	3	1	0	30	70	100
3.	MCS: 303	Cloud Computing	3	1	0	30	70	100
4.	MCS: 304	PHP	3	1	0	30	70	100
5.	MCS: 305	Android Applications	3	1	0	30	70	100
6.	MCS: 351	PHP Lab	0	0	4	30	70	100
7.	MCS: 352	Android Applications Lab	0	0	4	30	70	100
8.	MCS: 353	Minor Project	0	0	4	60	140	200
			15	5	12			900

Course Code	**MCS: 301**
Course Title	**Artificial Intelligence**
Credits	4
Course Objective(s)	The objective of an Artificial Intelligence (AI) course is to equip students with a comprehensive understanding of the core concepts, principles, and methodologies of AI. The course aims to introduce students to problem-solving techniques such as search algorithms, optimization methods, and heuristic approaches. It emphasizes the importance of knowledge representation, reasoning, and intelligent agent design, enabling students to develop systems capable of autonomous decision-making. Fundamental machine learning concepts, including supervised, unsupervised, and reinforcement learning, are explored to provide practical skills in building AI models. Additionally, the course delves into specialized areas such as Natural Language Processing (NLP) and Computer Vision, showcasing AI's applications across domains. Ethical considerations, including fairness, bias, and privacy, are integral to the curriculum, ensuring students develop responsible AI systems. Hands-on experience with modern tools like Python, Tensor Flow, and PyTorch prepares students for real-world challenges. By fostering research, innovation, and interdisciplinary applications, the course prepares students to address the evolving demands of AI-driven industries and contribute to technological advancements.
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: Natural Language Processing (NLP):** Provide an overview of NLP, text processing, and language models. Discuss techniques for sentiment analysis, machine translation, and chat bots.

Course Code	MCS: 302

MCS: 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, syntactic processing, 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. Clock sin, W.F and Mellish, C.S: Programming in PROLOG, Narosha Publishing

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. **CO2Mastering 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 apriority for practical problem-solving. **C06**: **Real-World Applications**: Understand how data warehousing and mining are applied in domains such as marketing, healthcare, and finance.

MCS: 302 Data warehousing & Data Mining

Course Code	MCS: 303

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 warehousing 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. Multi relational OLAP characteristics of OLAP, feature of OLAP, OLAP operation categorization of OLAP tool.

Main Text Books:

Jiawei Han & micheline Kamber: “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 Title	Cloud Computing
Credits	4
Course Objective(s)	A cloud computing course equips students with essential knowledge of service models (IaaS, PaaS, and SaaS) and deployment strategies (public, private, hybrid). Through hands-on assignments, students learn to deploy and manage cloud services effectively while developing analytical skills for evaluating solutions and ensuring security. The course emphasizes collaboration and culminates in assessments that demonstrate students' ability to implement cloud solutions, preparing them for careers in the field.
Course Outcome (CO)	**CO1: Understanding Cloud Computing Fundamentals:-** Learn the basic concepts, architecture, and delivery models (IaaS, PaaS, SaaS) of cloud computing. **CO2: Exploring Cloud Deployment Models:** Understand public, private, hybrid, and community cloud deployment models and their use cases. **CO3: Proficiency in Virtualization:** Gain knowledge of virtualization techniques and their role in enabling cloud services. **CO4: Cloud Service Management:** Learn about resource provisioning, load balancing, and auto-scaling in the cloud environment. **CO5: Knowledge of Cloud Platforms:** Explore popular cloud platforms like AWS, Microsoft Azure, and Google Cloud and their key features. **CO6: Data Management in the Cloud:** Understand cloud-based storage systems, data migration, and distributed data processing. **CO7: Cloud Security and Privacy:** Address security challenges, data protection, and compliance in cloud computing.

Course Code	MCS: 304

MCS: 303 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. Prasanta Pattnaik, Manas Kadat, Souvik Pal, Fundamentals of Cloud Computing.

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

3. Dr. Kumar Saurabh, Cloud Computing

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. **CO6: Build and Deploy Applications:** Understand the full lifecycle of web application development, from coding to deployment on web servers

Course Code	MCS: 305

MCS: 304 PHP

Unit -1 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 -2 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 Else If, Switch/Case Statement
1.12 PHP Operators: Logical, Relational, Bitwise, Ternary Operator (?)

Unit-3Arrays 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-4 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-5Databases 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 Title	Android Applications
Credits	4
Course Objective(s)	Upon completing the Android Applications course, students will be proficient in designing, building, and deploying mobile applications using Java or Kotlin in Android Studio. They will learn to create user interfaces, manage app resources, and utilize Android APIs for various functionalities. The course will cover best practices in app architecture, performance optimization, and security. Through hands-on projects, students will gain practical experience in the app development lifecycle, preparing them for careers in mobile application development.
Course Outcome (CO)	**CO1: Understand Android Architecture:** Gain a foundational understanding of the Android operating system and its architecture. **CO2: Proficient in Programming Languages:** Develop proficiency in Java or Kotlin for Android app development. **CO3: User Interface Design:** Learn to design and implement user-friendly interfaces using XML and Android UI components. **CO4: API Integration:** Understand how to integrate various Android APIs for functionalities such as location services, multimedia, and networking. **CO5: Data Management:** Learn to manage data using SQLite databases and shared preferences. **CO6: App Lifecycle Management:** Understand the Android application lifecycle and how to manage app states effectively..

MCS: 305 Android Applications

UNIT-1

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-2

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-3

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

UNIT-4

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, Grid View, Web View, Scroll View, Search View

UNIT-5

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	**MCS: 351**
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: 352**
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 Alarm Manager 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	**MCS-353**
Course Title	**Minor Project**
Credits	2
Course Objective(s)	The **Minor 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.

Pacific University, Udaipur
Faculty of Computer Application
Msc.IT (Master of Computer Science)- 2Yr Course
SEMESTER-IV

S. No.	Code	Subject Name	Period			Evaluation Scheme		
			L	T	P	ISE	ESE	Total
1.	MCS:401	Computer Based Numerical; and Statistical Techniques	3	1	0	30	70	100
2.	MCS:402	Computer Graphics	3	1	0	30	70	100
3.	MCS:403	Compiler Design	3	1	0	30	70	100
4.	MCS:404	Software Engineering	3	1	0	30	70	100
5.	MCS:451	CONUM Lab	0	0	4	30	70	100
6.	MCS:452	Graphics Lab	0	0	4	30	70	100
7.	MCS:453	Minor Project	0	0	4	150	250	400
			12	4	12			1000

Course Code	**MCS: 401**
Course Title	**Computer Based Numerical & Statistical Techniques**
Credits	4
Course Objective(s)	Upon completing the "Computer Based Numerical & Statistical Techniques" course, students will be proficient in applying key numerical and statistical methods using computer software. They will develop critical thinking and problem-solving skills, enabling them to analyze and interpret data effectively. Graduates will be prepared to tackle real-world problems in various fields, enhancing their ability to make informed decisions and contribute to collaborative research and professional projects
Course Outcome (CO)	**CO1**: **Understanding Numerical Methods**: To provide students with a solid foundation in numerical methods, including interpolation, numerical integration, differentiation, and solving ordinary differential equations. .. **CO2: Statistical Analysis:** To introduce students to fundamental statistical concepts and techniques, including descriptive statistics, inferential statistics, hypothesis testing, regression analysis, and analysis of variance (ANOVA). **CO3**: **Software Proficiency:** To develop proficiency in using computer software tools (such as Python, R, MATLAB, or Excel) for implementing numerical algorithms and performing statistical analyses. **CO4: Problem-Solving Skills:** To enhance students' problem-solving skills by applying numerical and statistical techniques to real-world problems across various fields such as engineering, finance, and social sciences. **CO5**: **Data Interpretation:** To train students in interpreting and visualizing data effectively, enabling them to draw meaningful conclusions and make informed decisions based on statistical analysis.

MCS-401: Computer Based Numerical & Statistical Techniques

UNIT-I

Floating point Arithmetic: Representation of floating point numbers, Operations, Normalization, Pitfalls of floating pointer presentation, Errors in numerical computation Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method.

UNIT-II

Regula- Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods.

UNIT-III

Interpolation and approximation: Newton's forward and backward Formula Langrange's Interpolation.

UNIT-IV

Differentiation: First derivative of principle, Runga- Kutta methods.

UNIT-V

Method of least squares, fitting of straight lines.

References:

1. Rajaraman V., "Computer Oriented Numerical Methods", PHI
2. Gerald & Wheatley, "Applied Numerical Analyses", AW
3. Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
4. Grewal B. S., "Numerical methods in Engineering and Science", Khanna Publishers, Delhi
5. T. Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods", TMH
6. PradipNiyogi, "Numerical Analysis and Algorithms", TMH
7. Francis Scheld, "Numerical Analysis", TMH
9. Gupta S. P., "Statistical Methods", Sultan and Sons

Course Code	**MCS: 402**
Course Title	**Computer Graphics**
Credits	4
Course Objective(s)	Upon completing the "Computer Graphics" course, students will understand key principles of 2D and 3D graphics, be proficient in graphics programming, and have hands-on experience with modelling and animation techniques. They will be equipped to design interactive applications, optimize performance for real-time graphics, and explore current trends like virtual and augmented reality, preparing them for careers in game development and visual effects.
Course Outcome (CO)	**CO1**: **Fundamentals of Computer Graphics:** To provide a comprehensive understanding of the basic principles and concepts of computer graphics, including 2D and 3D graphics, rendering techniques, and graphical transformations. **CO2: Mathematical Foundations:** To equip students with the necessary mathematical tools, such as linear algebra and geometry, that are essential for understanding and implementing graphics algorithms. **CO3**: **Graphics Programming:** To develop proficiency in graphics programming using relevant APIs and frameworks (such as OpenGL, DirectX, or WebGL) for creating interactive graphics applications. . **CO4: Rendering Techniques:** To explore various rendering techniques, including rasterization, ray tracing, and shading models, enabling students to create realistic images and animations. **CO5**: **Modelling and Animation**: To introduce students to 3D modelling and animation techniques, including the creation of geometric models, texture mapping, and key frame animation.

MCS-402 Computer Graphics

UNIT -I

A Survey of Computer Graphics

Application of graphics, Computer aided design, presentation graphics, computer Art, Entertainment, Education and training, Visualization, image Processing, GUI. input devices.

UNIT-II

Overview of computer graphics

Video display devices, refresh cathode ray tube, raster scan displays, random scan displays, color CRT displays, direct view storage tubes, flat panel displays, three dimensional viewing devices, raster scan systems, video controller, raster scan display processor, random scan systems.

UNIT-III

Graphics Objects

Points and lines, line drawing algorithms ,DDA algorithms, Brenham's line algorithm, circle generating algorithms, Filled area primitives, scan line polygon fill algorithm, scan line fill of curved boundary areas ,boundary fill algorithm, flood fill algorithm, fill area functions, character generation.

UNIT-IV

Two Dimensional Geometric Transformations.

Basic transformations Translation, rotation, scaling, Homogeneous Coordinates for translation, Homogeneous Coordinates for rotation, Homogeneous Coordinates for scaling, Reflection , Shear.

UNIT-V

2-D Viewing- The viewing pipeline. Viewing co-ordinate, Reference Frame. Windows to view ports .co-ordinate transformation 2-D

Viewing functions. Clipping operations point clipping.
Line clipping. Cohen- Sutherland. Line Clipping Polygon clipping. Sutherland Hodge man clipping.

Course Code	**MCS: 403**
Course Title	**Compiler Design**
Credits	4
Course Objective(s)	Upon completing the "Compiler Design" course, students will understand key concepts in compiler construction, including lexical analysis, parsing, semantic analysis, and code generation. They will gain hands-on experience by implementing a simple compiler, applying optimization techniques, and addressing error handling. This knowledge will prepare them for careers in software development and programming language design..
Course Outcome (CO)	**CO1**: **Understanding Compiler Structure:** To provide a comprehensive overview of the components and architecture of compilers, including lexical analysis, syntax analysis, semantic analysis, optimization, and code generation. **CO2: Lexical Analysis:** To teach students how to design and implement lexical analyzers (scanners) that convert source code into tokens. **CO3**: **Syntax Analysis:** To introduce parsing techniques and algorithms, enabling students to construct syntax trees and understand context-free grammars. **CO4: Semantic Analysis:** To cover the principles of semantic analysis, including type checking and scope resolution, ensuring that the code adheres to the language's rules. **CO5**: **Intermediate Representation:** To familiarize students with various intermediate representations of code and their role in optimization and code generation

MCS-403: Compiler Design:

UNIT I

Introduction: Analysis of source program, Different phases of a compiler, Symbol Table.

UNIT II

Lexical Analysis: Different approaches to design a lexical analyzer, regular expression, finite automata (Deterministic & Non-deterministic). RE to NFA and NFA to DFA. Optimization of DFA states. Implementation of lexical analyzer.

UNIT III

Error Handling: errors in different phases of compiler. Introduction to Compiler Construction Tools.

Syntax analysis: context free grammar, parsing techniques (Top-down, Bottom-up, Operator-precedence, SLR, LALR).

UNIT IV

Intermediate code generation: Intermediate language, syntax directed translation, assignment statement, Boolean statements and back patching, array references, procedure calls and record structure.

UNIT V

Code optimization: Principal sources of optimization, Local & Loop optimization, loop invariant computations, induction variable elimination.

Code generation: Design of code generation, a machine model, a simple code generator, register allocation & assignment, code generation from DAG's.

Course Code	**MCS: 404**
Course Title	**Software Engineering**
Credits	4
Course Objective(s)	Upon completing the "Software Engineering" course, students will understand the software development life cycle, be proficient in requirements gathering, design, and testing, and apply Agile methodologies. They will gain skills in project management and collaboration tools, ensuring high-quality software delivery while being aware of ethical considerations and current industry trends.
Course Outcome (CO)	**CO1**: **Understanding Software Development Life Cycle (SDLC):** To provide a comprehensive overview of the various phases of software development, including requirements gathering, design, implementation, testing, deployment, and maintenance. **CO2: Requirements Engineering:** To teach students how to effectively gather, analyze, and document software requirements, ensuring that the final product meets user needs. **CO3**: **Software Design Principles:** To introduce students to design methodologies and principles, including architectural patterns, design patterns, and user interface design. **CO4: Project Management**: To cover project management techniques, including planning, scheduling, risk management, and resource allocation, to ensure successful project delivery. **CO5**: **Quality Assurance and Testing:** To emphasize the importance of software testing and quality assurance practices, including unit testing, integration testing, and system testing. **CO6**: **Version Control and Collaboration**: To teach the use of version control systems and collaborative tools that facilitate teamwork in software development. **CO7**: **Software Maintenance and Evolution**: To discuss strategies for maintaining and evolving software systems over time, addressing issues such as technical debt and refactoring.

MCS-404: Software Engineering

UNIT-I

Introduction to Software Engineering:-Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Software Myth. Similarity and Difference from Conventional Engineering Processes, Software Quality Attributes.

Software Development Life Cycle (SDLC) Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Model.

UNIT-II

Software Requirement Specification (SRS):-Requirement Engineering Process: Elicitation, Analysis, Documentation Review and Management of User Needs, Feasibility Study, Information Modelling, Data Flow Diagrams, Entity Relationship Diagrams, Decision, Tables, SRS Document, IEEE Standard for SRS.

Software Quality Assurance (SQA):- Verification and Validation, SQA Plan, Software Quality Framework, ISO 9000 Model, SEI-CMM Model.

UNIT-III

Software Design:- Basic Concept of Software Engineering, Architectural Design, Low Level Design: Modularization, Design Structure Chart, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom Up Design, Software Measurement and Matrices.

Function Point (FP) Based Measures, Cyclamate Complexity Measure: Control Flow Graphs.

Software Testing:-Testing Ojectives,UnitTesting,IntegrationTesting,AcceptanceTesting,Regression Testing for Functionality, and Test for Performing, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stub, Structural Testing(White Box Testing),Functional Testing(Black Box Testing),Alpha and Beta Testing of Products.

UNIT-IV

Software Maintenance and Software Project Management:-Software as an Evolutionary Entity, Need for Maintenance, Category of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-engineering, Reverse Re-Engineering.

Software Configuration Management Activities: Change Control Process, Software Version Control, An Overview of CASE Tolls.

UNIT-V

Estimation Of Various Parameter:- Such As Cost,Efforts,Schedule/Duration, Constrictive Cost Model(COCOMO),Resource Allocation. Models, Software Risk Analysis and Management.

CASE(Computer Aided Software Engineering):- CASE and its Scope, CASE support in Software Life Cycle, Documentation Project Management, Internal Interface, Reverse Software Engineering, Architecture of CASE Environment.

Reference Book:-

1.Pressman, RogerS.,"Software Engineering a Partioner Approach edBoston:McGraw Hill,2001

2. Jalote, Pankaj,"Software EngineeringEd.2",New Delhi:Narosa2002,Schaum's Series, "Software Engineering",TMH,Ghezzi,Carol and Others,

3. "Fundamentals of Software Engineering", PHI

Course Code	**MCS: 451**
Course Title	**COUNM LAB**
Credits	2
Course Objective(s)	The course on "Computer-Based Numerical and Statistical Methods" enables students to effectively apply numerical algorithms and statistical techniques using computational tools, enhancing their problem-solving skills and data analysis capabilities for real-world applications.
Course Outcome (CO)	**CO1**: **Practical Application of Theories:** To provide students with hands-on experience in applying numerical and statistical methods learned in theory to solve real-world problems using computer software. **CO2: Familiarity with Software Tools:** To familiarize students with various computational tools and programming languages (such as Python, R, MATLAB, or others) that are commonly used for numerical analysis and statistical computations. **CO3**: **Implementation of Algorithms**: To enable students to implement and test various numerical algorithms (e.g., root-finding, interpolation, numerical integration) and statistical techniques (e.g., regression analysis, hypothesis testing) in a computational environment. **CO4: Data Analysis Skills:** To develop students' skills in data collection, processing, and analysis, allowing them to draw meaningful insights from datasets using statistical methods. **CO5**: **Error Analysis and Debugging:** Error Analysis and Debugging: To teach students how to identify, analyze, and mitigate errors in numerical computations and statistical analyses, enhancing their problem-solving skills.

List of Lab Programs:

1. 1 Numerical Solution of tri-diagonal system using Thomas algorithm
2. 2 Solution of simultaneous non-linear equations using Newton Raphsonmethod.
3. Solution of simultaneous non-linear equations using Newton Raphson Method(Two variables)
4. Numerical solution of central difference interpolation by using Lagrange interpolation method
5. Numerical solution of integrals using Trapezoidal method
6. Numerical evaluation of integrals using Simpson's method.
7. Numerical solution of characteristic value problem by Power method.
8. Numerical Solution of initial value problem using Euler's method
9. Numerical Solution of initial value problem using Runga Kutta method
10. 10 Numerical Solution for heat equation by suitable method

Course Code	**MCS: 452**
Course Title	**Graphic LAB**
Credits	2
Course Objective(s)	Upon completing the Graphic Lab course, students will demonstrate a solid understanding of graphic design principles and techniques, showcasing their ability to create effective visual communications across various media. They will gain proficiency in industry-standard software, apply critical thinking to design challenges, and collaborate effectively in team settings. Students will also develop a professional portfolio that highlights their skills and creativity, preparing them for success in the graphic design industry while adhering to ethical standards and staying informed about current trends.
Course Outcome (CO)	**CO1** : **Understanding Graphic Design Principles**: Learn the fundamental principles of graphic design, including colour theory, typography, layout, and composition. **CO2: Proficiency in Graphic Software:** Gain hands-on experience with industry-standard graphic design software (e.g., Adobe Photoshop, Illustrator, and InDesign) and understand their tools and functionalities. **CO3**: **Creative Problem Solving:** Develop the ability to approach design challenges creatively and effectively, applying design thinking methodologies. **CO4: Project Development:** Work on individual and group projects that involve conceptualizing, designing, and presenting graphic materials for various purposes (e.g., branding, advertising, digital media). **CO5**: **Portfolio Development:** Create a professional portfolio showcasing a range of graphic design projects, demonstrating skills and creativity to potential employers or clients.

List of Lab Programs:

1. Create a promotional poster for a fictional event. What design principles did you use?
2. Develop a logo for a new start-up company. How does it reflect the brand's values?
3. Choose a quote and design a typographic poster. How do your choices enhance its impact?
4. Create a surreal image by combining multiple photographs in Photoshop. What techniques did you use?
5. Design an info graphic that visually represents a set of data. How do you ensure clarity and visual hierarchy?
6. Create a brand identity package that includes a logo, colour palette, and business card design. How does each element reflect the brand's identity?
7. Design a homepage mock-up for a website. What layout and navigation choices did you make for user experience?
8. Develop a series of three cohesive social media graphics for a marketing campaign. How do they align with the campaign message?
9. Present a design project and lead a critique session. What challenges did you face, and how did you address feedback?
10. Create a colour palette for a specific project. How do your choices relate to colour theory principles?

Course Code	**MCS: 453**
Course Title	**Minor Project**
Credits	2
Course Objective(s)	The **Minor 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 analysing 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.