Re-Accredited 'B++' 2.86 CGPA by NAAC.
VEER NARMAD SOUTH GUJARAT UNIVERSITY
University Campus, Udhna-Magdalla Road, SURAT - 395 007, Gujarat, India.
વીર નર્મદ દક્ષિણ ગુજરાત યુનિવર્સિટી
યુનિવર્સિટી કેમ્પસ, ઉધના-મગદલ્લા રોડ, સુરત - ૩૯૫ ૦૦૭, ગુજરાત, ભારત.
Tel : +91 - 261 - 2227141 to 2227146, Toll Free : 1800 2333 011, Digital Helpline No.- 0261 2388888
E-mail : info@vnsgu.ac.in, Website : www.vnsgu.ac.in

ક્રમાંક :ઓથો./પરિપત્ર/૧૨૨૬૫/૨૦૨૫
તા.૨૦/૦૫/૨૦૨૫

પ્રતિ,
વડાશ્રી,
કોમ્પ્યુટર સાયન્સ વિભાગ,
વીર નર્મદ દક્ષિણ ગુજરાત યુનિવર્સિટી,
સુરત.

વિષયઃ– Master of Computer Application (MCA), Sem-3 & 4 નો અભ્યાસક્રમ અંગે.

સુજ્ઞશ્રી,

સવિનય જણાવવાનું કે, શૈક્ષણિક વર્ષ ૨૦૨૫–૨૬ થી અમલમાં આવનાર પેટાસમિતિ દ્વારા તૈયાર કરવામાં આવેલ Master of Computer Application (MCA), Sem-3 & 4 નો અભ્યાસક્રમ અંગે કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા તથા કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની સંયુકત તા.૩૦/૦૪/૨૦૨૫ ની સભાનાં ઠરાવ ક્રમાંકઃ ૧૩ થી કરેલ ભલામણ સ્વીકારી એકેડેમિક કાઉન્સિલની તા.૦૫/૦૫/૨૦૨૫ની સભાનાં ઠરાવ ક્રમાંકઃ ૧૫૮ થી મંજુર કરેલ છે. જેનો અમલ કરવા આથી જાણ કરવામાં આવે છે.

બિડાણઃ ઉપર મુજબ

કુલસચિવ

પ્રતિ,
૧) ઈ.ચા.ડીનશ્રી, કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા.
૨) પરીક્ષા નિયામકશ્રી, પરીક્ષા વિભાગ, વીર નર્મદ દ. ગુ. યુનિવર્સિટી, સુરત.
.....તરફ જાણ તેમજ અમલ સારૂ.

Master of Computer Applications

Name of Programme	**Master of Computer Applications**
Abbreviation	**MCA**
Duration	**2 Years**
Eligibility Criteria	Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree. OR Passed B.Sc./ B.Com./ B.A. with Mathematics at 10+2 Level or at Graduation Level (with additional bridge Courses as per the norms of the concerned University).
Objective of Programme	The core objective of the MCA programme is to prepare the students for productive career in software industry and academia by providing an outstanding environment of teaching and research in the core and emerging areas of the discipline.
Program Outcomes	PO1: Fundamental Knowledge Enrichment Program trains students with the core computer science and Information Technology (IT) knowledge domains. It also makes students capable of using core concepts in the conceptualization of domain specific application development. PO2: Critical Thinking Development The program develops the skills of critical thinking, problem solving, evaluative learning of various techniques, and understanding the essence of the problem. PO3: Advanced Emerging Technology Awareness The program trains students with the latest technologies that is being used in the industry. The continuous syllabi review adds value to the program for the outgoing students and make them ready to face challenging demands of the industry. PO4: Advanced Tools Usage The program teaches the students to apply the advanced tools to solve real world problems. PO5: Nurturing Project Planning and Management Capabilities The program trains students for designing and conceptualizing the software architecture, planning and managing the product development process of complex and live software projects. It also makes students understand the decision making for selection of an appropriate project management capabilities. PO6: Real World Problem / Project Development Real world project provides the candidates exposure to work in the challenging and demanding environment of the industry. The project development training makes students employable and industry ready. PO7: Team Work and Leadership Development Trains students to work in a team and also to take leadership of the of the project management team.
Program Specific Outcomes	PSO1: Develop and strengthen the fundamental core concepts that are required to solve complex problems PSO2: Develop the professional and entrepreneurship skills that needs independent logical and analytical thinking, teamwork and leadership PSO3: Nurture the students to investigate for the design and development of a workable solution for a real world problem PSO4: Develop students for self-learning and practicing challenging problem solution

	PSO5: Train students to apply managerial skills to develop business applications. PSO6: Train students to use recent computer science and application domain specific knowledge PSO7: Train students to take-up the real world challenges to develop workable solution to a domain specific problem PSO8: Inculcate the passion for continuous learning and doing research for making a successful professional career

Mapping between POs and PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	
PO1									
PO2									
PO3									
PO4									
PO5									
PO6									
PO7									

Medium of Instruction	English
Programme Structure	**Semester 3 (Web Group)**

Course Code	Title	Teaching per week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
301	Embedded Systems and IOT Machine Learning	4	0	4	3 Hrs.	70	30	100
302	Design Patterns	4	0	4	3 Hrs.	70	30	100
303	Advanced Web Technologies Advanced Java Programming	4	0	4	3 Hrs.	70	30	100
304	Full Stack Technology	4	0	4	3 Hrs.	70	30	100
305	Open Source Web Based Programming	4	0	4	3 Hrs.	70	30	100
306	Programming Skills VIII	0	2	2	2 Hrs.	70	30	100
307	Programming Skills IX	0	3	3	2 Hrs.	70	30	100
308	Programming Skills X	0	2	2	2 Hrs.	70	30	100
309	Programming Skills XI	0	3	3	2 Hrs.	70	30	100
	Total	20	10	30	23 Hrs.	630	270	900

Programme Structure	**Semester 3 (Database Group)**

Course Code	Title	Teaching per Week		Course Credits	University Examination		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
301	Embedded Systems and IOT Machine Learning	4	0	4	3 Hrs.	70	30	100
302	Design Patterns	4	0	4	3 Hrs.	70	30	100
303	ERP Using SAP NoSQL Databases	4	0	4	3 Hrs.	70	30	100
304	Advanced Database Administration	4	0	4	3 Hrs.	70	30	100
305	Data Warehousing and Data Mining Big Data	4	0	4	3 Hrs.	70	30	100
306	Programming Skills VIII	0	2	2	2 Hrs.	70	30	100
307	Programming Skills IX	0	3	3	2 Hrs.	70	30	100
308	Programming Skills X	0	2	2	2 Hrs.	70	30	100
309	Programming Skills XI	0	3	3	2 Hrs.	70	30	100
	Total	20	10	30	23 Hrs.	630	270	900

Programme Structure	**Semester 3 (Network Group)**

Course Code	Title	Teaching per week		Course Credits	University Exam		Internal Marks	Total Marks
		Theory	Practical		Duration	Marks		
301	Embedded Systems and IOT Machine Learning	4	0	4	3 Hrs.	70	30	100

302	Design Patterns	4	0	4	3 Hrs.	70	30	100
303	Network Essential & its Security	4	0	4	3 Hrs.	70	30	100
304	Network Administration	4	0	4	3 Hrs.	70	30	100
305	Wireless Network & Mobile Computing	4	0	4	3 Hrs.	70	30	100
306	Programming Skills VIII	0	2	2	2 Hrs.	70	30	100
307	Programming Skills IX	0	3	3	2 Hrs.	70	30	100
308	Programming Skills X	0	2	2	2 Hrs.	70	30	100
309	Programming Skills XI	0	3	3	2 Hrs.	70	30	100
	Total	20	10	30	23 Hrs.	630	270	900

Programme Structure | **Semester 3 (General Group)**

Course Code	Title	Teaching per week		Course Credits	University Exam		Internal Exam	Total Marks
		Theory	Practical		Duration	Marks		
301	Embedded Systems and IOT Machine Learning	4	0	4	3 Hrs.	70	30	100
302	Design Patterns	4	0	4	3 Hrs.	70	30	100
303	Network Essential & its Security	4	0	4	3 Hrs.	70	30	100
304	Advanced Database Administration	4	0	4	3 Hrs.	70	30	100
305	Open Source Web Based Programming	4	0	4	3 Hrs.	70	30	100
306	Programming Skills VIII	0	2	2	2 Hrs.	70	30	100
307	Programming Skills IX	0	3	3	2 Hrs.	70	30	100
308	Programming Skills X	0	2	2	2 Hrs.	70	30	100
309	Programming Skills XI	0	3	3	2 Hrs.	70	30	100
	Total	20	10	30	23 Hrs.	630	270	900

Programme Structure | **Semester 4**

Course Code	Title	Course Credit	University Exam Marks	Internal Marks	Total Marks
401	Seminar	6	70	30	100
402	Project	24	280	120	400

MCA 3rd Sem. (Web Group)

[Subject code-2111040103010002]

Course: 301: Embedded Systems and IOT

(Elective)

Course Code	301
Course Title	**Embedded Systems and IOT**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course is an introduction for students to embedded system and IOT. The course also gives students an idea about various components of microcontroller, microprocessor and IOT and explain its working. The course also explains the role of embedded systems in IOT ecosystem.
Course Objective	The objective of the course is to: 1. Understand the concept of embedded systems, including their internal architecture and programming of embedded processors. 2. Apply the knowledge of interfacing techniques to connect input/output devices with processors. 3. Understand the fundamentals of the Internet of Things (IOT), including its architecture and key components. 4. Analyze the architecture and operational features of microcontrollers and microcomputers. 5. Identify and classify various types of sensors used in embedded and IOT systems. 6. Develop and Implement basic IOT programs using suitable embedded platforms.
Course Outcome	CO1 (Remembering / Understanding): Recall and explain the fundamentals of 8-bit microcontrollers, including their architecture, instruction sets, and interfacing with peripherals. CO2 (Applying): Apply Embedded C programming techniques for memory interfacing, I/O devices, and serial communication in embedded systems. CO3 (Understanding): Understand and describe the role of Real-Time Operating Systems (RTOS) in embedded systems, including task management and priority-based scheduling. CO4 (Analyzing): Analyze the working principles and features of microcontroller platforms like Arduino and NodeMCU (ESP8266), and their use in embedded system design. CO5 (Evaluating): Evaluate and select appropriate communication protocols (Wi-Fi, Bluetooth, ZigBee, GSM) for IoT systems based on application requirements. CO6 (Creating): Design and implement IoT applications such as smart agriculture, home automation, and smart healthcare, using sensors, embedded systems, and communication protocols.
Mapping between COs with PSOs	
Pre-requisite	Computer Architecture, Programming in C, Programming with C++
Course Content	**Unit 1: 8-Bit Embedded Processor** 1.1 8-Bit Microcontroller 1.2 Architecture 1.3 Instruction Set and Programming 1.4 Programming Parallel Ports 1.5 Timers and Serial Port 1.6 Interrupt Handling. **Unit 2: Embedded C Programming**

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1								
CO2								
CO3								
CO4								
CO5								

	2.1 Memory And I/O Devices Interfacing 2.2 Programming Embedded Systems in C 2.3 Need For RTOS 2.4 Multiple Tasks and Processes 2.5 Context Switching 2.6 Priority Based Scheduling Policies. **Unit 3: IOT and Arduino Programming** 3.1 Introduction to Microcontrollers: Architecture, Features, and Programming Basics 3.2 Arduino: Development and Programming 3.3 Arduino IDE Setup 3.4 Arduino Architecture 3.5 Arduino Pin Diagram 3.6 Introduction to NodeMCU (ESP8266) 3.7 NodeMCU Specifications and Applications 3.8 NodeMCU ESP8266 Pinout **Unit 4: IOT Sensors, Communication and open platforms and Applications development** 4.1 Introduction to Various Sensors 4.2 Interfacing Sensors with Arduino and Node MCU 4.3 IOT Communication Models and APIs 4.4 IOT Communication Protocols – Bluetooth – WiFi – ZigBee– GPS – GSM modules 4.5 Introduction to raspberry Pi 4.6 Architecture of Raspberry Pi 4.7 Complete Design of Embedded Systems 4.8 Development of IOT Applications: Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare
Reference Books	1. Getting Started with Internet of Things – By Cuno Pfister, O'Reilly 2. Learning Internet of Things – By Peter Waher , Packt Publication 3. Internet of Things: A Hands-on Approach – By Arshdip Bahga and Vijay Madisetti 4. IOT Governance, Privacy and Security Issues, IERC 5. IOT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things, Cisco Press 6. Fundamentals of IOT Communication Technologies, Springer 7. Microcontrollers – Architecture, Programming, Interfacing and system design – By Raj Kamal , Pearson 8. Exploring C for Microcontrollers: A hands on approach, Springer 9. Arduino for Dummies, Wiley 10. Make: Getting Started With Arduino - The Open Source Electronics Prototyping Platform, Shroff/Maker Media 11. ESP8266: Get Started With ESP8266 Programming NodeMCU Using Arduino IDE, Createspace Independent Pub 12. Internet of Things Projects with ESP32, Packt Publishing Limited 13. Microprocessor Architecture, Programming and Applications with the 8085 - By Ramesh Gaonkar , Penram International Publishing 14. Raspberry Pi for Dummies , Wiley 15. Raspberry Pi User Guide – By Eben Upton and Garath Halfacree, Wiley
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103010003]

Course: 301: Machine Learning

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Machine Learning (ML)</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to ML. The course also gives students an idea about various methods and algorithms of Machine Learning and application development of ML.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is –
1. To make student understand ML
2. To understand the various Machine Learning method
3. To explain various algorithms used in Machine learning
4. To introduce students with Programming in ML</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Define Machine Learning and its types (Supervised, Unsupervised, and Semi-supervised) and identify the tools and applications associated with it.
CO2 (Applying): Apply data preprocessing techniques, select appropriate models, and train models using tools like Scikit-learn.
CO3 (Understanding): Explain the importance of data wrangling, including data cleaning, transformation, reduction, discretization, and feature selection, and utilize data visualization techniques.
CO4 (Analyzing): Analyze and differentiate between supervised learning algorithms, including regression (Simple, Multiple, Linear, Gradient Descent, Logistic) and classification algorithms (KNN, SVM, Decision Trees, Naïve Bayes).
CO5 (Applying): Implement clustering algorithms (K-means, Hierarchical) and association rule learning algorithms (FP-Growth, Apriori) for unsupervised learning tasks.
CO6 (Creating): Design and implement neural networks, including their architecture, feedforward networks, and backpropagation, for solving real-world problems.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Basics of Linear Algebra, Statistics and Mathematics, Python Programming</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 Definition of Machine Learning and types of Machine Learning: Supervised, Unsupervised and Semi-supervised
1.2 Applications and tools of Machine Learning (Scikit learn library)
1.3 Data Pre-processing, Selecting a model and training a model
1.4 Evaluating a performance of model and improving performance
1.5 Data Wrangling
1.5.1 Definition and goal of Data Wrangling
1.5.2 Importance of Data Wrangling
1.5.3 Data Pre-processing and Data Cleaning
1.5.3.1 Data Cleaning
1.5.3.2 Data Transformation
1.5.3.3 Data Reduction
1.5.3.4 Data Discretization</td></tr>
</table>

	1.5.3.5 Feature Selection 1.5.4 Data Visualization **Unit 2: Supervised Learning** 2.1 Supervised Learning: Classification and Regression 2.2 Regression 2.2.1 Simple and Multiple Regression 2.2.2 Linear Regression 2.2.3 Gradient Decent 2.2.4 Logistic Regression 2.3 Classification Algorithms : 2.3.1 K-nearest Neighbor 2.3.2 Support Vector Machines 2.3.3 Decision Trees 2.3.4 Naïve Bayes Classifier 2.4 Introduction to Support Vector Machine **Unit 3: Unsupervised Learning** 3.1 Introduction to Unsupervised learning 3.2 Clustering 3.2.1 Selection of Clusters 3.2.2 Algorithms: 3.2.2.1 K – means clustering 3.2.2.2 Hierarchical Clustering 3.3 Association Rule Learning 3.3.1 Algorithms : 3.3.1.1 FP- Growth 3.3.1.2 Apriori Algorithm **Unit 4: Neural Network** 4.1 Introduction to Neural Network 4.2 Architecture of Neural Network 4.3 Feedforward network and Backpropagation with example 4.4 Applications of Neural Network
Reference Books	1. “Machine Learning” by Tom M. Mitchell, McGraw Hill 2. “Understanding Machine Learning” by Shai Shalev-Shwartz, Shai Ben-David 3. “Machine Learning” by Anuradha Srinivasaraghavan, Vincy Joseph 4. “Machine Learning using Python” by U Dinesh Kumar Manaranjan Pradhan 5. “Real-World Machine Learning” by Henrik Brink, Joseph Richards, Mark Fetherolf 6. “Python Machine Learning” by Sebastian Raschka and Vahid Mirjalili 7. “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems” by Aurelien Geron 8. “Machine Learning in Action” by Peter Harrington 9. “Introduction to Machine Learning with Python : A Guide for Data Scientists” by Andreas C. Muller, Sarah Guido
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103020002]

Course: 302: Design Patterns

<table>
<tr><td>Course Code</td><td>302</td></tr>
<tr><td>Course Title</td><td>Design Patterns</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>The purpose of the course is to make student understand how Patterns can be implemented in various object oriented programming languages to solve real world problems.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is -
1. To study various Design Patterns
2. How these Patterns can be used to design better systems through Object Oriented Programming Languages</td></tr>
<tr><td>Course Outcome</td><td>CO1: Understand the purpose and use of design patterns and remember the SOLID patterns which are the building block for design patterns. Remember the various design patterns; their categories, and standard patterns as per GOF.
CO2: Analyze the first category of creational design patterns and learn them.
CO3: Understand the structural design patterns.
CO4: Understand and analyze the behavioral design patterns.
CO5: Learn some more design patterns used in IT industry currently.
CO6: Make students understand the applicability of design patterns practiced by IT companies and how effectively combine these patterns for effective software development.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Object Oriented Programming, Software Engineering</td></tr>
<tr><td>Course Content</td><td>Unit -1: Introduction to Design Patterns
1.1 What’s a Design Pattern?
1.2 Benefits of Design Patterns
1.3 Software Design Principles
1.4 SOLID Principles
1.5 Design patterns – their categories and standard design patterns as per GOF

Unit-2: Creational and Structural Patterns
2.1 Creational Patterns
2.1.1 Factory
2.1.2 Abstract Factory
2.1.3 Builder
2.1.4 Prototype
2.1.5 Singleton
2.2 Structural Patterns
2.2.1 Adapter
2.2.2 Bridge
2.2.3 Composite
2.2.4 Decorator
2.2.5 Overview of other Structural Patterns - Façade, Flyweight and Proxy patterns

Unit-3: Behavioral Patterns
3.1 Chain of Responsibility
3.2 Command
3.3 Iterator</td></tr>
</table>

	3.4 Mediator 3.5 Overview of other Behavioral Patterns - Memento, Observer, State, Strategy, Template Method and Visitor Patterns **Unit 4: Additional Patterns and Patterns Applicability** 4.1 Overview of Additional Patterns - Simple Factory, Null Object and MVC Patterns 4.2 Security Patterns Repository 4.3 Patterns for Agile Development 4.4 Relation between patterns and pattern combinations Self-Study : Implementation of Patterns in various Programming Languages
Reference Books	1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph, John, Addision Wesley 2. Drive into Design Patterns, Alexander Shvets, Refactoring.Guru, 2019 3. Head First Design Patterns, Eric Freeman, O'Reilly 4. Design Patterns in C#, Vaskaran Sarcar, Apress 5. Design Patterns in Modern C++, Reusable Approaches for Object-Oriented Software Design, Dmitri Nesteruk, Apress 6. Modern C++ design: generic Programming and design patterns applied, Alexendrescu, Andrei, Addison-Wesley 7. Java Design Patterns: A Hands-on Experience with Real-World Examples, Vaskaran Sarcar, Apress
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103030002]

Course: 303: Advanced Web Technologies

(Elective)

Course Code	303
Course Title	**Advanced Web Technologies**
Credit	4
Teaching per Week	4
Minimum weeks per Semester	15 (Including Classwork, examination, preparation, holidays etc.)
Effective From	June 2025
Purpose of Course	The purpose of the course is to make the student understand and implement the Concepts of Advanced web applications development and server side development using .NET framework.
Course Objective	To impart knowledge of web application development and backend development using ASP.NET, MVC & .NET 8
Course Outcomes	CO1 (Remember & Understand): Explain the fundamental aspects of the .NET framework and ASP.NET, identifying key concepts and principles. CO2 (Apply & Analyze): Utilize various tools and controls available in ASP.NET for web application development, integrating them effectively to build cohesive applications. CO3 (Apply & Evaluate): Implement database operations using Entity Framework, comparing design alternatives and coding approaches, and critically analysing the application of Language Integrated Query (LINQ). CO4 (Apply & Evaluate): Develop applications using the MVC architecture to meet modern development requirements, incorporating component-based development APIs and assessing their efficiency. CO5 (Understand & Analyze): Compare and contrast .NET Core and .NET 8 with previous versions, identifying advancements in .NET architecture and evaluating its impact on application development. CO6 (Apply & Create): Design and implement web-based applications utilizing the latest tools and techniques in ASP.NET, integrating server-side component development for optimal performance and functionality.
Mapping between COs with PSOs	
Pre-requisite	Fundamental of .NET framework and HTML desirable.
Course Content	**Unit 1 : Introduction to .NET Frameworks** 1.1 Overview of .NET Framework 1.2 .NET Core 1.2.1 Overview of .NET Core 1.2.2 .NET Core Assemblies and Libraries 1.2.3 Features of .NET Core 1.3 .NET 8 1.3.1 Evolution of .NET and its Significance 1.3.2 Key enhancements and new features in .NET 8 1.3.3 Advantages of .NET 8 in modern development 1.4 Overview of C# language enhancements in .NET 8 1.4.1 Primary Constructor 1.4.2 Default Parameters in Lambda Expression 1.4.3 Interceptors 1.4.4 Improved Pattern Matching **Unit 2 : Introduction to ASP.NET** 2.1 Overview of ASP.NET

	PSO1	PSO2	PSO3	POS4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■		■					
CO3	■	■	■			■		
CO4		■	■	■	■	■	■	■
CO5		■	■	■	■	■	■	■
CO6	■	■		■	■		■	■

	2.2 Page Class and Page Life Cycle 2.3 Web Configuration files 2.4 Exception Handling 2.5 Error Pages 2.6 ASP.NET State Management 2.7 Introduction to Caching **Unit 3 : ASP .NET - MVC & Data Access** 3.1 ASP.NET MVC 3.1.1 MVC Architecture 3.1.2 URL Routing Engine 3.1.3 Writing Controller 3.1.4 Writing Model 3.1.5 Customizing View with Razor 3.2 LINQ 3.2.1 LINQ language features 3.2.2 LINQ to Objects 3.2.3 LINQ to SQL 3.3 Entity Framework 3.3.1 Code First Approach 3.3.2 Data First Approach **Unit 4 : Advancements in ASP.NET Development** 4.1 API 4.1.1 RESTful API 4.1.2 Introduction to JSON 4.1.3 API Creation and Consumption 4.2 Blazor Framework 4.2.1 Introduction to Blazor (Server and Web Assembly) 4.2.2 Integrating with .NET 4.2.3 Building interactive and reusable components 4.2.4 Dependency Injection 4.2.5 State management in Blazor 4.2.6 Integrating APIs with Blazor Applications 4.3. Advanced Topics in .NET 8 4.3.1 Minimal APIs in .NET 8 4.3.2 Performance Enhancement and Tuning 4.3.3 Asynchronous Programming with .NET 4.4 Cloud-native Development with .NET 4.4.1 Introduction to Docker 4.4.2 Introduction to Kubernetes
Reference Books	1. Professional ASP.NET, Wrox Publication 2. ASP.NET – From Novice to Professional, Wrox Publication 3. ASP.NET Bible, By Mridula Parihar 4. Beginning ASP.NET 4.5, Wrox Publication 5. Programming Microsoft ASP.NET, Microsoft Press 6. Professional ASP.NET MVC 5, Wrox Publication 7. Professional C# 7 and .NET Core 2.0 , Wrox Publication 8. ASP.NET Core 2 Fundamentals, Packt Publication 9. Pro ASP.NET MVC 5, Apress 10. Programming ASP.NET Core, Microsoft Press 11. Pro C# 7 with .NET and .NET Core, Apress 12. C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals, Packt 13. Architecting ASP.NET Core Applications - Third Edition: An atypical design patterns guide for .NET 8, C# 12, and beyond, packt 14. Apps and Services with .NET 8, Packt 15. Professional C# and .NET, 2021st Edition, Wrox

Teaching Methodology	Classwork, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103030003]

Course: 303: Advanced Java Programming

(Elective)

Course Code	303
Course Title	**Advanced Java Programming**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Effective From	June 2025
Purpose of Course	This course is advance level java course to learn web enabled application development using Java Technologies.
Course Objective	1. To develop skills in using web-based Java technologies. 2. To design and build Java web applications using server-side technologies.
Course Outcomes	CO1 (Remembering / Understanding): Define JDBC concepts, types of drivers, and demonstrate connections to databases such as MySQL, Oracle, SQL Server, and Access. CO2 (Applying): Work with JDBC objects such as Connection, ResultSet, Statement, Metadata, and utilize JDBC connection pooling for efficient database interactions. CO3 (Applying): Implement Java Servlets to handle HTTP requests, manage sessions, use servlet annotations, filters, and perform file uploads. CO4 (Analyzing): Analyze the structure and lifecycle of JSP, use JSTL and Expression Language (EL) to manage session tracking and work with database tags in JSP applications. CO5 (Creating): Develop web services using JAX-WS and JAX-RS, and integrate Java Persistence API (JPA) to manage database operations with persistent query language (JPQL). CO6 (Understanding / Applying): Explain and implement Enterprise Java Beans (EJB) for session management and explore the MVC framework along with Spring architecture for building robust web applications.
Mapping between COs with PSOs	
Pre-requisite	Core Java, Object oriented Programming
Course Content	**Unit 1: JDBC (Java Database Connectivity)** 1.1 Types of JDBC Drivers 1.2 Connecting to databases such as Access, MySQL, SqlServer, Oracle 1.3 Working with JDBC Objects: Connection, Resultset, Statement, Metadata 1.4 More JDBC Objects: DataSource, RowSet 1.5 Calling Stored Procedures 1.6 Transactions management & Concurrency control 1.7 JDBC Connection Pooling **Unit 2. Java Servlets** 2.1 Introduction to Servlets & Servlet Lifecycle 2.2 Handling HTTP GET and POST requests 2.3 Invoking other web resources 2.4 Session management

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■		■					
CO3	■	■			■			
CO4		■	■	■		■		
CO5	■		■		■			
CO6	■	■		■		■	■	■

	2.5 Servlet Annotations 2.6 Servlet Filter 2.7 File Upload **Unit 3: JSP, JSTL & EL** 3.1 Introduction to JSP & JSP lifecycle 3.2 JSP Elements – Expressions, Directives, Declarations, Scriptlets 3.3 JSP actions & Implicit JSP objects 3.4 Session Tracking in JSP 3.5 Using JavaBeans in JSP 3.6 JSTL – Using Java Standard Tag Library 3.7 JSTL Core & Database tags 3.8 Introduction to EL (Expression Language) 3.9 EL implicit objects **Unit 4: Web Services, JPA and EJB** 4.1 Web Services 4.1.1 Introduction to JAXP and JAXB 4.1.2 Building XML based web services with JAX-WS 4.1.3 Building Restful web services with JAX-RS 4.2 Introduction to Java Persistence API (JPA) 4.2.1 Java Persistent Unit, Entity Manager 4.2.2 Persistent query Language (JPQL) 4.3 Enterprise Java Bean 4.3.1 Introduction to EJB & EJB container 4.3.2 Session Bean 4.4 Overview of MVC Framework & Spring Architecture
Reference Books	1. "Java: The Complete Reference" by Herbert Schildt (McGraw-Hill Education), Ninth Edition 2. Java EE Tutorial Basic Concepts by Oracle Corporation 3. Beginning Java™ EE Platform with GlassFish™ : From Novice to Professional by Antonio Goncalves 4. Beginning EJB 3 Application Development From Novice to Professional by Raghu R.Kodali and Jonathan Wetherbee with Peter Zadrozny, Apress Publication 5. Pro JPA 2: Mastering the Java™ Persistence API 6. Head First Servlets and JSP By: Bryan Basham, Kathy Sierra, Bert Bates Publisher: 'Reilly Media 7. Core Servlets and Javaserver Pages: Author Marty Hall , Larry Brown, Sun Micro System 8. Java Servlet & JSP Cookbook by Bruce W. Perry O;reilly Publication 9. Beginning JSP™, JSF™ and Tomcat™ Web Development: From Novice to Professional by Giulio Zambon and Michael Sekler 10. Professional Java Development with Spring Framework , Wrox Publication 11. Restful Java with Jax-RS 2.0: Designing and Developing Distributed Web Services, Bill Burke A I , O'reilly Publication
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103040002]

Course: 304: Full Stack Technology

Course Code	304
Course Title	**Full Stack Technology**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	The purpose of the course is to make the students capable of developing full stack web applications.
Course Objective	The objective of the course is to make student understand full stack development
Course Outcome	CO1 (Remembering / Understanding): Identify key concepts and features of JavaScript, including execution context, hoisting, closures, ES6+ features, and the basic principles of full stack development, including the MERN stack. CO2 (Applying): Implement JavaScript concepts like the event loop, promises, async/await, and perform data storage and retrieval using localStorage, sessionStorage, and cookies. CO3 (Applying): Develop dynamic, interactive web applications using React.js components, state management (useState, useEffect, Redux), and handle forms, events, and API interactions. CO4 (Analyzing): Analyze and apply advanced React concepts, including routing, real-time data with WebSockets, performance optimization techniques (memoization, lazy loading), and security practices like authentication strategies. CO5 (Creating): Create and deploy full-stack web applications by integrating React.js with Node.js and Express.js, developing RESTful APIs, handling authentication and security, and deploying applications to cloud platforms. CO6 (Applying / Creating): Utilize MongoDB for managing data in a full-stack MERN application, implementing CRUD operations, schema definitions, and optimization techniques such as aggregation, transactions, and performance tuning.
Mapping between COs with PSOs	
Pre-requisite	HTML, CSS, Front-end Scripting
Course Content	**Unit 1: JavaScript & Full Stack Technology** 1.1 JavaScript Essentials 1.1.1 Execution Context and Call Stack 1.1.2 Hoisting in JavaScript 1.1.3 Spread and Rest Operators 1.1.4 Scope Chain, Temporal Dead Zone 1.1.5 Block Scope, Shadowing 1.1.6 Closures and Lexical Scope 1.1.7 ES6+ Features (let, const, arrow functions, destructuring, modules) 1.2 Advanced JavaScript Concepts 1.2.1 Event Loop and Asynchronous JavaScript 1.2.2 Promises, Async/Await 1.2.3 Callbacks and Callback Hell 1.2.4 Fetch API and Error Handling 1.2.5 Working with LocalStorage, SessionStorage, Cookies 1.3 Full Stack Technology Overview 1.3.1 Introduction to Full Stack Development 1.3.2 Understanding Frontend, Backend, and Databases 1.3.3 Introduction to MERN Stack (MongoDB, Express.js, React.js, Node.js)

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1								
CO2								
CO3								
CO4								
CO5								

Unit 2: React.js

2.1 Introduction to React.js
- 2.1.1 What is React?
- 2.1.2 Key Features of React
- 2.1.3 JSX and Virtual DOM

2.2 Setting Up React
- 2.2.1 Installing React (npx create-react-app)
- 2.2.2 Understanding React Project Structure

2.3 React Components and Advanced Concepts
- 2.3.1 Functional vs. Class Components
- 2.3.2 Parent-Child Communication
- 2.3.3 Conditional Rendering
- 2.3.4 Props and Prop Types
- 2.3.5 Higher-Order Components (HOCs)
- 2.3.6 Render Props Pattern

2.4 Event Handling & Forms in React
- 2.4.1 Event Handling (onClick, onChange, etc.)
- 2.4.2 Handling Forms and Validations
- 2.4.3 Controlled vs. Uncontrolled Components
- 2.4.4 Libraries: react-hook-form, formik, yup

2.5 State Management & Hooks
- 2.5.1 useState, useEffect, useRef, useReducer
- 2.5.2 useContext for Global State Management
- 2.5.3 Advanced State Management using Redux Toolkit
- 2.5.4 Server State Management with React Query

2.6 Routing with React Router
- 2.6.1 React Router Basics (Link, Navigate)
- 2.6.2 Nested and Dynamic Routes
- 2.6.3 Route Guards and Protected Routes

2.7 Performance Optimization in React
- 2.7.1 Memoization (useMemo, useCallback)
- 2.7.2 Debouncing & Throttling
- 2.7.3 Lazy Loading and Code Splitting (React.lazy, Suspense)

2.8 API Handling and Real-time Data
- 2.8.1 Fetch API and Axios
- 2.8.2 Handling API Errors and Loading States
- 2.8.3 WebSockets and Real-time Data

2.9 Authentication & Security
- 2.9.1 Authentication Strategies (JWT, OAuth, Firebase Auth)
- 2.9.2 Role-based Authorization
- 2.9.3 Secure API Calls (httpOnly Cookies vs. LocalStorage)

2.10 Deployment of React Apps
- 2.10.1 Preparing React for Production (build process)
- 2.10.2 Deploying to Vercel, Netlify, Firebase
- 2.10.3 SEO Optimization (react-helmet, next-seo)

Unit 3: Backend Development with Node.js & Express.js and Full Stack Integration & Deployment

3.1 Introduction to Node.js
- 3.1.1 What is Node.js?
- 3.1.2 Understanding the Event Loop
- 3.1.3 Creating a Simple Node.js Server

3.2 Express.js Framework
- 3.2.1 Introduction to Express.js
- 3.2.2 Creating a Simple Express Server
- 3.2.3 Handling Requests and Responses
- 3.2.4 Express Middleware (Built-in & Custom Middleware)

	3.2.5 Error Handling in Express 3.3 API Development with Express.js 3.3.1 RESTful API Concepts 3.3.2 CRUD Operations with Express.js 3.3.3 Express Router and Route Parameters 3.4 Authentication & Security 3.4.1 JWT Authentication 3.4.2 OAuth and Google/Facebook Login 3.4.3 Secure API Endpoints 3.5 Connecting React with Node.js & Express.js 3.5.1 Setting Up CORS for API Access 3.5.2 Integrating Frontend and Backend 3.6 Building a Full Stack MERN Application 3.6.1 User Authentication (Login/Signup) 3.6.2 Managing User Roles (Admin/User) 3.6.3 CRUD Operations in a MERN App 3.6.4 Real-time Features (WebSockets, Push Notifications) **Unit 4: MongoDB Database** 4.1 Introduction to MongoDB 4.1.1 NoSQL vs. SQL Databases 4.1.2 MongoDB Features and Use Cases 4.2 MongoDB with Node.js 4.2.1 Connecting Node.js to MongoDB (mongoose) 4.2.2 CRUD Operations in MongoDB 4.2.3 Schema and Model Definition 4.2.4 Data Validation and Indexing 4.3 Advanced MongoDB Concepts 4.3.1 Aggregation Framework 4.3.2 Transactions in MongoDB 4.3.3 Optimizing Queries and Performance
Reference Books	1. Node.js for Beginners: A Comprehensive Guide to Building Efficient and Scalable Applications, Author: John Smith, Publisher: TechPress, ISBN: 978-1803245174, Year: 2024. 2. Mastering Node.js, Express & MongoDB Development: 2024 Guide for Beginners, Author: Jane Doe, Publisher: CodeMaster Publishing, ISBN: 978-1803245181, Year: 2024. 3. Modern Full-Stack React Projects, Author: Mary Johnson, Publisher: Packt Publishing, ISBN: 978-1837637959, Year: 2023. 4. Learning React: A Hands-On Guide to Building Web Applications Using React and Redux (3rd Edition), Author: Kirupa Chinnathambi, Publisher: Addison-Wesley Professional, ISBN: 978-0137647281, Year: 2023. 5. JavaScript: The Definitive Guide (7th Edition), Author: David Flanagan, Publisher: O'Reilly Media, ISBN: 978-1491952023, Year: 2023. 6. Eloquent JavaScript (4th Edition), Author: Marijn Haverbeke, Publisher: No Starch Press, ISBN: 978-1718502713, Year: 2023. 7. Full-Stack React Projects (2nd Edition), Author: Shama Hoque, Publisher: Packt Publishing, ISBN: 978-1801070631, Year: 2022. 8. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node (2nd Edition), Author: Vasan Subramanian, Publisher: Apress, ISBN: 978-1484243909, Year: 2022. 9. Beginning Node.js, Express & MongoDB Development, Author: Greg Lim, Publisher: Independently Published, ISBN: 978-1678123216, Year: 2022. 10. MongoDB: The Definitive Guide (4th Edition), Author: Shannon Bradshaw, Publisher: O'Reilly Media, ISBN: 978-1491954461, Year: 2022.
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103050002]

Course: 305: Open Source Web Based Programming

Course Code	305
Course Title	**Open Source Web Based Programming**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to understand fundamentals of Open-Source web-based Programming. The course also imparts students learning about Open-source web-based scripting language PHP and MySQL database. It also includes MVC or Three tier architecture of web-based programming.
Course Objective	1. Identify and recall the fundamental concepts, tools, and terminologies used in open-source web technologies. 2. Explain the architecture and working principles of various open-source web technologies and frameworks. 3. Develop basic web applications using open-source tools and integrate essential components like forms, databases, and APIs. 4. Differentiate between various open-source frameworks (e.g., ReactJS, Node.js, etc.) and analyze their advantages and limitations in specific use cases. 5. Design and build a fully functional, database-driven web application using modern open-source technologies, adhering to best practices in structure, security, and UI/UX.
Course Outcome	CO1 (Remembering / Understanding): Recognize the fundamentals of PHP and MySQL, their installation and configuration, and identify language features such as operators, variables, control structures, and error handling. CO2 (Applying): Apply PHP functions to perform string manipulation, array operations, mathematical calculations, and handle date/time operations, while also using object-oriented features like classes, objects, constructors, and inheritance. CO3 (Applying): Establish and manage MySQL databases, execute CRUD operations, handle server-side communication using PHP and AJAX, and implement form validation and real-time user feedback in dynamic web applications. CO4 (Analyzing): Analyze and integrate XML for data representation and web services for remote communication, utilizing AJAX for asynchronous requests and dynamic content handling in web applications. CO5 (Applying / Creating): Develop full-stack web applications using the Laravel framework by applying MVC architecture, managing routing, controllers, and views, handling form submissions, and ensuring application security with features like CSRF, XSS, and SQL injection prevention. CO6 (Applying / Creating): Implement advanced concepts in Laravel such as dynamic UI rendering with Blade, email communication, user authentication, role-based access control, and data encryption, while ensuring secure data handling and applying best practices in application security.
Mapping between COs with PSOs	
Pre-requisite	Knowledge of HTML and SQL
Course Content	**Unit 1 : Introduction to Open source Web based Programming** 1.1 Introduction to PHP & MySql 1.2 Installation of PHP and MySql 1.3 Language Characteristics & Features

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■		■					
CO3	■	■	■		■			
CO4		■	■	■		■	■	■
CO5	■		■		■	■		
CO6	■	■		■		■	■	■

	1.4 Operators and Variables, Control Structures, Looping and Error handling 1.5 PHP functions 1.5.1 String Functions 1.5.2 Array Functions 1.5.3 Mathematical Functions 1.5.4 Graphics Library (GD Support) 1.5.6 Date and Time Functions 1.5.7 Misc. Function 1.6 State management Techniques 1.7 Object Oriented Features of PHP 1.7.1 Classes and Objects 1.7.2 Use of constructors 1.7.3 Serialization 1.7.4 Inheritance **Unit 2: Advanced Server-Side Integration and Dynamic Web Communication** 2.1 MySQL Server Setup and Management 2.1.1. Installation and Configuration of MySQL Server 2.1.2. Creating and Managing Tables 2.1.3. Displaying Databases and Managing User Access 2.1.4. Establishing Database Connections and Data Handling using Server-Side Scripts 2.2 AJAX-Based Web Communication 2.2.1 Fundamentals of HTTP Requests and Responses 2.2.2 The XMLHttpRequest Object: Methods and Properties 2.2.3 Handling Cross-Browser AJAX Implementations 2.2.4 Sending Asynchronous Requests to the Server 2.2.5 Integrating PHP with AJAX for Client-Server Communication 2.2.6 Server-Side Processing and Dynamic Content Handling 2.2.7 Form Validation and Real-Time User Feedback 2.2.8 Executing AJAX-Driven Database Queries 2.3 Structured Data Exchange and Services 2.3.1 Introduction to XML for Data Representation 2.3.2 Basics of Web Services for Remote Communication and Integration **Unit 3: Full-Stack Web Development with Laravel Framework** 3.1 MVC Architecture & Laravel Introduction 3.1.1 Fundamentals of MVC (Model-View-Controller) Architecture 3.1.2 Overview of Laravel and its Core Concepts 3.2 Application Structure & Routing 3.2.1 Laravel Application Workflow 3.2.2 Route Definitions and URL Mapping 3.2.3 Middleware for Request Filtering 3.3 Request Lifecycle & Controllers 3.3.1 Handling HTTP Requests and Responses 3.3.2 Creating and Managing Controllers 3.4 Views & Dynamic Content Rendering 3.4.1 Blade Templating Engine for View Management 3.4.2 URL Generation and Layout Inheritance 3.5 Form Handling & User Input 3.5.1 Form Submission and File Upload Mechanisms 3.5.2 Input Validation Techniques 3.5.3 Session Management and Cookie Usage 3.6 Error Management & Application Security 3.6.1 Exception Handling and Custom Error Pages 3.6.2 Security Features in Laravel (CSRF, XSS, SQL Injection Prevention) 3.7 Database Integration 3.7.1 Connecting to Databases Using Eloquent ORM 3.7.2 Performing CRUD Operations and Data Relationships

	Unit 4 : Advanced Concepts in MVC with Laravel 4.1 Dynamic UI Rendering with Blade 4.1.1 Advanced Blade Templating Techniques 4.1.2 Layout Inheritance, Components, and Slots 4.1.3 Conditional and Loop Directives 4.2 Email Communication in Laravel 4.2.1 Sending Emails using Laravel Mail 4.2.2 Email Templates and Queueing Mails 4.2.3 SMTP Configuration and Mail Drivers 4.3 User Authentication 4.3.1 Built-in Authentication Scaffolding 4.3.2 Login, Registration, and Password Reset 4.3.3 Customizing Authentication Logic 4.4 Access Control and Authorization 4.4.1 Role-Based Access Control (RBAC) 4.4.2 Gates and Policies for Fine-Grained Access 4.4.3 Middleware for Route Protection 4.5 Data Security and Encryption 4.5.1 Encrypting and Decrypting Data 4.5.2 Storing Secure Information 4.5.3 Laravel's Cryptographic Tools and Best Practices
Reference Books	1. Beginning PHP, Apache, MySQL Web Development - Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, Gary Mailer Wrox Publication 2. Professional PHP Programming - Jesus Castagnetto ,Wrox Press Ltd 3. Beginning PHP and MySQL: From Novice to Professional - W. Jason Gilmore, Apress 4. Php: The Complete Reference - Steven Holzner, Tata Mcgraw Hill Education Private Limited 5. AJAX and PHP: Building Responsive Web Applications - Bogdan Brinzarea, CristianDarie packtpub 6. Php manual www. Php.com 7. Beginning Laravel Build Websites with Laravel 5.8 - Sanjib Sinha · 2019 - Apress 8. Laravel The Ultimate Beginner's Guide to Learn Laravel Step by Step, 2nd Edition - Mem Lnc, Rufus Stewart
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-*2111040103060002*]

Course: 306: Programming Skills VIII

Course Code	306
Course Title	**Programming Skills VIII**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn practically Embedded systems and IOT / ML
Course Objective	Learn Embedded systems and IOT practically, understand the working of Micro-Controller & Micro-Computer and using various types of sensors and its Programming. / Implement various algorithms used in Machine learning and introduce students with Programming in ML.
Pre-requisite	C/C++ Language / Python Programming
Course Outcome	After successful completion, students will be able to work with different types of Micro-Controllers, Micro-Computers and sensors for their IOT based application development. / After successful completion, students will be able to work with different types of ML algorithms and ML based application development.
Course Content	Practical based on paper no 301. (Embedded systems and IOT/ML) Separate journal to be prepared for this Course
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises , Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-*2111040103070002*]

Course: 307: Programming Skills IX

Course Code	307
Course Title	**Programming Skills IX**
Credit	3
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to implement the advanced concepts of .NET/Java practically.
Course Objective	Learn how to implement the advanced topics of .NET/Java practically.
Pre-requisite	Practical programming in basic .NET/Java.
Course Outcome	After studying the course, students will be able to practically work on advanced technology platforms of .NET/Java.
Course Content	Practical based on paper no 303. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code- 2111040103080002]

Course: 308: Programming Skills X

Course Code	308
Course Title	**Programming Skills X**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn how to develop full stack web applications.
Course Objective	Learn to implement full stack development
Pre-requisite	HTML, CSS, Front-end Scripting
Course Outcome	After completion of this course, the students will be able to design and develop Full Stack web applications
Course Content	Practical based on paper no 304. (Full Stack Technology) Separate journal to be prepared for this Course
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103090002**]**

Course: 309: Programming Skills XI

Course Code	309
Course Title	**Programming Skills XI**
Credit	3
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn how to implement the basic and advanced concepts of PHP-MySql practically.
Course Objective	Learn to develop and deploy websites using PHP-MySql practically.
Pre-requisite	Basic scripting, programming, html.
Course Outcome	After studying the course, students will be able to practically develop dynamic websites using PHP- MySql.
Course Content	Practical based on paper no 305. (Open Source Web Based Programming) Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

MCA 3rd Sem

(Database Group)

[[[[**[Subject code-2111040103010002]**

Course: 301: Embedded Systems and IOT

(Elective)

Course Code	301
Course Title	**Embedded Systems and IOT**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course is an introduction for students to embedded system and IOT. The course also gives students an idea about various components of microcontroller, microprocessor and IOT and explain its working. The course also explains the role of embedded systems in IOT ecosystem.
Course Objective	The objective of the course is to: 1. Understand the concept of embedded systems, including their internal architecture and programming of embedded processors. 2. Apply the knowledge of interfacing techniques to connect input/output devices with processors. 3. Understand the fundamentals of the Internet of Things (IOT), including its architecture and key components. 4. Analyze the architecture and operational features of microcontrollers and microcomputers. 5. Identify and classify various types of sensors used in embedded and IOT systems. 6. Develop and Implement basic IOT programs using suitable embedded platforms.
Course Outcome	CO1 (Remembering / Understanding): Recall and explain the fundamentals of 8-bit microcontrollers, including their architecture, instruction sets, and interfacing with peripherals. CO2 (Applying): Apply Embedded C programming techniques for memory interfacing, I/O devices, and serial communication in embedded systems. CO3 (Understanding): Understand and describe the role of Real-Time Operating Systems (RTOS) in embedded systems, including task management and priority-based scheduling. CO4 (Analyzing): Analyze the working principles and features of microcontroller platforms like Arduino and NodeMCU (ESP8266), and their use in embedded system design. CO5 (Evaluating): Evaluate and select appropriate communication protocols (Wi-Fi, Bluetooth, ZigBee, GSM) for IoT systems based on application requirements. CO6 (Creating): Design and implement IoT applications such as smart agriculture, home automation, and smart healthcare, using sensors, embedded systems, and communication protocols.
Mapping between COs with PSOs	
Pre-requisite	Computer Architecture, Programming in C, Programming with C++
Course Content	**Unit 1: 8-Bit Embedded Processor** 1.1 8-Bit Microcontroller

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1								
CO2								
CO3								
CO4								
CO5								
CO6								

	1.2 Architecture 1.3 Instruction Set and Programming 1.4 Programming Parallel Ports 1.5 Timers and Serial Port 1.6 Interrupt Handling. **Unit 2: Embedded C Programming** 2.1 Memory And I/O Devices Interfacing 2.2 Programming Embedded Systems in C 2.3 Need For RTOS 2.4 Multiple Tasks and Processes 2.5 Context Switching 2.6 Priority Based Scheduling Policies. **Unit 3: IOT and Arduino Programming** 3.1 Introduction to Microcontrollers: Architecture, Features, and Programming Basics 3.2 Arduino: Development and Programming 3.3 Arduino IDE Setup 3.4 Arduino Architecture 3.5 Arduino Pin Diagram 3.6 Introduction to NodeMCU (ESP8266) 3.7 NodeMCU Specifications and Applications 3.8 NodeMCU ESP8266 Pinout **Unit 4: IOT Sensors, Communication and open platforms and Applications development** 4.1 Introduction to Various Sensors 4.2 Interfacing Sensors with Arduino and Node MCU 4.3 IOT Communication Models and APIs 4.4 IOT Communication Protocols – Bluetooth – WiFi – ZigBee– GPS – GSM modules 4.5 Introduction to raspberry Pi 4.6 Architecture of Raspberry Pi 4.7 Complete Design of Embedded Systems 4.8 Development of IOT Applications: Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare
Reference Books	1. Getting Started with Internet of Things – By Cuno Pfister, O'Reilly 2. Learning Internet of Things – By Peter Waher , Packt Publication 3. Internet of Things: A Hands-on Approach – By Arshdip Bahga and Vijay Madisetti 4. IOT Governance, Privacy and Security Issues, IERC 5. IOT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things, Cisco Press 6. Fundamentals of IOT Communication Technologies, Springer 7. Microcontrollers – Architecture, Programming, Interfacing and system design – By Raj Kamal , Pearson 8. Exploring C for Microcontrollers: A hands on approach, Springer 9. Arduino for Dummies, Wiley 10. Make: Getting Started With Arduino - The Open Source Electronics Prototyping Platform, Shroff/Maker Media 11. ESP8266: Get Started With ESP8266 Programming NodeMCU Using Arduino IDE, Createspace Independent Pub 12. Internet of Things Projects with ESP32, Packt Publishing Limited 13. Microprocessor Architecture, Programming and Applications with the 8085 - By Ramesh Gaonkar , Penram International Publishing 14. Raspberry Pi for Dummies , Wiley 15. Raspberry Pi User Guide – By Eben Upton and Garath Halfacree, Wiley

Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103010003]

Course: 301: Machine Learning

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Machine Learning (ML)</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to ML. The course also gives students an idea about various methods and algorithms of Machine Learning and application development of ML.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is –
1. To make student understand ML
2. To understand the various Machine Learning method
3. To explain various algorithms used in Machine learning
4. To introduce students with Programming in ML</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Define Machine Learning and its types (Supervised, Unsupervised, and Semi-supervised) and identify the tools and applications associated with it.
CO2 (Applying): Apply data preprocessing techniques, select appropriate models, and train models using tools like Scikit-learn.
CO3 (Understanding): Explain the importance of data wrangling, including data cleaning, transformation, reduction, discretization, and feature selection, and utilize data visualization techniques.
CO4 (Analyzing): Analyze and differentiate between supervised learning algorithms, including regression (Simple, Multiple, Linear, Gradient Descent, Logistic) and classification algorithms (KNN, SVM, Decision Trees, Naïve Bayes).
CO5 (Applying): Implement clustering algorithms (K-means, Hierarchical) and association rule learning algorithms (FP-Growth, Apriori) for unsupervised learning tasks.
CO6 (Creating): Design and implement neural networks, including their architecture, feedforward networks, and backpropagation, for solving real-world problems.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Basics of Linear Algebra, Statistics and Mathematics, Python Programming</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 Definition of Machine Learning and types of Machine Learning: Supervised, Unsupervised and Semi-supervised
1.2 Applications and tools of Machine Learning (Scikit learn library)
1.3 Data Pre-processing, Selecting a model and training a model
1.4 Evaluating a performance of model and improving performance
1.5 Data Wrangling
1.5.1 Definition and goal of Data Wrangling
1.5.2 Importance of Data Wrangling
1.5.3 Data Pre-processing and Data Cleaning
1.5.3.1 Data Cleaning
1.5.3.2 Data Transformation</td></tr>
</table>

	1.5.3.3 Data Reduction 1.5.3.4 Data Discretization 1.5.3.5 Feature Selection 1.5.4 Data Visualization **Unit 2: Supervised Learning** 2.1 Supervised Learning: Classification and Regression 2.2 Regression 2.2.1 Simple and Multiple Regression 2.2.2 Linear Regression 2.2.3 Gradient Decent 2.2.4 Logistic Regression 2.3 Classification Algorithms : 2.3.1 K-nearest Neighbor 2.3.2 Support Vector Machines 2.3.3 Decision Trees 2.3.4 Naïve Bayes Classifier 2.4 Introduction to Support Vector Machine **Unit 3: Unsupervised Learning** 3.1 Introduction to Unsupervised learning 3.2 Clustering 3.2.1 Selection of Clusters 3.2.2 Algorithms: 3.2.2.1 K – means clustering 3.2.2.2 Hierarchical Clustering 3.3 Association Rule Learning 3.3.1 Algorithms : 3.3.1.1 FP- Growth 3.3.1.2 Apriori Algorithm **Unit 4: Neural Network** 4.1 Introduction to Neural Network 4.2 Architecture of Neural Network 4.3 Feedforward network and Backpropagation with example 4.4 Applications of Neural Network
Reference Books	1. “Machine Learning” by Tom M. Mitchell, McGraw Hill 2. “Understanding Machine Learning” by Shai Shalev-Shwartz, Shai Ben-David 3. “Machine Learning” by Anuradha Srinivasaraghavan, Vincy Joseph 4. “Machine Learning using Python” by U Dinesh Kumar Manaranjan Pradhan 5. “Real-World Machine Learning” by Henrik Brink, Joseph Richards, Mark Fetherolf 6. “Python Machine Learning” by Sebastian Raschka and Vahid Mirjalili 7. “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems” by Aurelien Geron 8. “Machine Learning in Action” by Peter Harrington 9. “Introduction to Machine Learning with Python : A Guide for Data Scientists” by Andreas C. Muller, Sarah Guido
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103020002]

Course: 302: Design Patterns

<table>
<tr><td>Course Code</td><td>302</td></tr>
<tr><td>Course Title</td><td>Design Patterns</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>The purpose of the course is to make student understand how Patterns can be implemented in various object oriented programming languages to solve real world problems.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is -
1. To study various Design Patterns
2. How these Patterns can be used to design better systems through Object Oriented Programming Languages</td></tr>
<tr><td>Course Outcome</td><td>CO1: Understand the purpose and use of design patterns and remember the SOLID patterns which are the building block for design patterns. Remember the various design patterns; their categories, and standard patterns as per GOF.
CO2: Analyze the first category of creational design patterns and learn them.
CO3: Understand the structural design patterns.
CO4: Understand and analyze the behavioral design patterns.
CO5: Learn some more design patterns used in IT industry currently.
CO6: Make students understand the applicability of design patterns practiced by IT companies and how effectively combine these patterns for effective software development.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Object Oriented Programming, Software Engineering</td></tr>
<tr><td>Course Content</td><td>Unit -1: Introduction to Design Patterns
1.1 What’s a Design Pattern?
1.2 Benefits of Design Patterns
1.3 Software Design Principles
1.4 SOLID Principles
1.5 Design patterns – their categories and standard design patterns as per GOF

Unit-2: Creational and Structural Patterns
2.1 Creational Patterns
2.1.1 Factory
2.1.2 Abstract Factory
2.1.3 Builder
2.1.4 Prototype
2.1.5 Singleton
2.2 Structural Patterns
2.2.1 Adapter
2.2.2 Bridge
2.2.3 Composite
2.2.4 Decorator
2.2.5 Overview of other Structural Patterns - Façade, Flyweight and Proxy patterns</td></tr>
</table>

	Unit-3: Behavioral Patterns 3.1 Chain of Responsibility 3.2 Command 3.3 Iterator 3.4 Mediator 3.5 Overview of other Behavioral Patterns - Memento, Observer, State, Strategy, Template Method and Visitor Patterns **Unit 4: Additional Patterns and Patterns Applicability** 4.1 Overview of Additional Patterns - Simple Factory, Null Object and MVC Patterns 4.2 Security Patterns Repository 4.3 Patterns for Agile Development 4.4 Relation between patterns and pattern combinations Self-Study : Implementation of Patterns in various Programming Languages
Reference Books	1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph, John, Addision Wesley 2. Drive into Design Patterns, Alexander Shvets, Refactoring.Guru, 2019 3. Head First Design Patterns, Eric Freeman, O'Reilly 4. Design Patterns in C#, Vaskaran Sarcar, Apress 5. Design Patterns in Modern C++, Reusable Approaches for Object-Oriented Software Design, Dmitri Nesteruk, Apress 6. Modern C++ design: generic Programming and design patterns applied, Alexendrescu, Andrei, Addison-Wesley 7. Java Design Patterns: A Hands-on Experience with Real-World Examples, Vaskaran Sarcar, Apress
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103030004]

Course: 303: ERP Using SAP

<table>
<tr><td>Course Code</td><td>303</td></tr>
<tr><td>Course Title</td><td>ERP Using SAP</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Last Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course imparts fundamental as well as detailed Enterprise Resource Planning using SAP technology platform.</td></tr>
<tr><td>Course Objective</td><td>Learn ERP applications and its programming</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Understand): Explain the fundamental aspects of Enterprise Resource Planning (ERP) and describe the role and significance of SAP in ERP systems.
CO2 (Apply): Demonstrate the ability to represent declarative knowledge related to the SAP Application Server and illustrate the structure of the 3-Tier Architecture.
CO3 (Understand & Apply):Interpret the concepts of Data Dictionary and apply data structures effectively in ABAP programming.
CO4 (Apply & Analyze): Implement Modularization Techniques, Module Pool Programming, Batch Data Communication (BDC), Legacy System Migration Workbench (LSMW), and Selection-Screen programming, and analyze their role in real-time SAP applications.
CO5 (Analyze & Evaluate):Analyze the end-to-end SAP implementation process and evaluate how ERP business processes are transformed to meet organizational needs.
CO6 (Apply & Create): Demonstrate the ability to work with various ERP modules and develop a basic ERP application using SAP tools and methodologies.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><td></td><td>PSO1</td><td>PSO2</td><td>PSO3</td><td>PSO4</td><td>PSO5</td><td>PSO6</td><td>PSO7</td><td>PSO8</td></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Fundamentals of ERP, DBMS</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction, Architecture of SAP Application server & Data Dictionary & Data Structures in ABAP
1.1 Introduction to ERP
1.2 Introduction to SAP
1.3 Example: How SAP works in an Organization
1.4 3-Tier Architecture
1.5 Application Servers
1.6 Work processes and its Type
1.7 Introduction to Data dictionary
1.8 Different Types of Data structures
1.9 Internal Tables and its operation

Unit 2: Modularization Techniques, List Report, ALV Report
2.1 Include Programs
2.2 Subroutines
2.3 Function Module.
2.4 Types of Function Module(Simple, RFC enabled, BAPI)
2.5 Simple List Report
2.6 Interactive List Report
2.7 Events in List Reports
2.8 Field catalog generation in ALV</td></tr>
</table>

	2.9 Operation on ALV(Sorting, Filtering, Totals, Subtotals, Download, Hide Columns) Unit 3: Module pool programming / Screen Programming & Selection-Screen Programming 3.1 Screen Elements(Simple & Complex) 3.2 Screen Events(PBO/PAI) 3.3 Transactions 3.4 Defining Selection Screen. 3.5 User Actions on Selection Screen. 3.6 Events of Selection Screen Unit 4: Smartform/Sapscript, BDC & LSMW & Enhancement(Exits & BADI) 4.1 Form printing with smartform 4.2 Form printing with Sapscript 4.3 Data upload through BDC 4.4 Data upload through LSMW 4.5 What is Enhancement 4.6 User-Exits 4.7 BADI(Business Add-in)
Reference Books	1. ABAP Cookbook by James Wood 2. BC - ABAP Programming from SAP-AG 3. Teach Yourself ABAP/4 in 21 Days by Ken Greenwood, SAMS 4. SAP Smart Forms by Christoph Wachter, Werner Hertleif 5. SAPscript by Michaelson Buchanan 6. Developing Sap's R/3 Application with Abap/4 7. Data Migration Made Easy - R/3 Simplications Group, SAP Labs 8. ABAP Development for SAP NetWeaver BW: Exits, BAdIs, and Enhancements by Dirk Herzog 9. Next Generation ABAP Development (2nd Edition) by Rich Heilman and Thomas Jung
Teaching Methodology	Discussion, Independent Study, Seminars and Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103030005]

Course: 303: NoSQL Databases

<table>
<tr><td>Course Code</td><td>303</td></tr>
<tr><td>Course Title</td><td>NoSQL Databases</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Last Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>To teach the emerging trends in NoSQL databases</td></tr>
<tr><td>Course Objective</td><td>To impart knowledge of NoSQL Databases</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remember & Understand): Describe the history, concept, characteristics, and types of NoSQL databases.
CO2 (Understand & Analyze): Compare NoSQL databases with relational databases, and explain their advantages, disadvantages, and real-world applications.
CO3 (Understand): Explain the fundamentals, feature set, and architecture of MongoDB.
CO4 (Apply): Demonstrate the ability to create documents, collections, and databases in MongoDB, and construct simple and complex queries for data insertion, updating, and retrieval.
CO5 (Apply & Analyze): Implement MongoDB RESTful API operations and apply security measures in MongoDB-based applications.
CO6 (Understand & Analyze): Explain the architecture and data modeling concepts of Cassandra and analyze how Cassandra integrates with Hadoop for big data solutions.

CO7 (Apply & Evaluate):
Create and manage keyspaces and databases in Cassandra, and demonstrate Cassandra administrative tasks for efficient database management.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Fundamentals of DBMS</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 History
1.2 Concepts and Characteristics of NoSQL databases
1.3 Primary benefits of NoSQL databases

Unit 2: MongoDB
2.1 SQL/NoSQL landscape
2.2 Document Vs. Other types of Storage
2.3 MongoDB feature set
2.4 Introduction to BSON and JSON
2.5 MongoDB Architecture
2.6 Documents and Collections
2.6.1 Creating Documents
2.6.2 Managing Documents in collections</td></tr>
</table>

	2.6.3 Iterating over Documents 2.7 Queries 2.7.1 Simple Queries 2.7.2 Complex Queries 2.7.2.1 Existential field values 2.7.2.2 Aggregations and groups 2.7.2.3 Aggregations and groups in hierarchical data 2.8 Updates and Deletes 2.9 Updates and Arrays 2.10 Indexing **Unit 3: MongoDB – RESTful API & Security** 3.1 MongoDB RESTful API 3.2 MongoDB Security 3.3 MongoDB Replication and Sharing 3.4 Introduction to MapReduce **Unit 4: Cassandra** 4.1 Cassandra Architecture 4.1.1 Cassandra P2P Architecture 4.1.2 Clustering Structures- Nodes 4.1.3 Rings 4.1.4 Virtual Nodes 4.1.5 Consistency & Hashing 4.1.6 Gossip Protocol 4.1.7 Data Replication 4.1.8 Replication Factors & Indexes 4.1.9 Tunable Consistency 4.1.10 High & Rapid Scalability Memtables, SStables & Commitlogs 4.1.11 Repairs 4.1.12 Tombstones 4.1.13 Repairs 4.1.14 Replication Factors 4.1.15 Compaction and Anti-Entropy 4.1.16 Bloom Filters 4.2 Data Modelling in Cassandra 4.3 Cassandra Administration 4.4 CQL3 4.5 Integration with Hadoop
Reference Books	1. Chodorow, K. (2013). MongoDB: The Definitive Guide (2nd ed.). Upper Saddle River, NJ: Pearson Education, Inc. ISBN-13: 978-1449344689 ISBN-10: 1449344682. 2. Shashank Tiwari, Professional NoSQL, Sierra Nevada Books, ISBN-13: 978-0470942246 3. Amol Nayak, Instant MongoDB, Packt Publishing Limited, 2013, ISBN-13**:** 978-1782169703 4. Kristina Chodorow, MongoDB Definitive Guide 2e, O'Reilly, 2013, ISBN-13: 978-1449344689 5. Eben Hewitt, Cassandra Definitive Guide, O'Reilly, 2010, ISBN:ISBN 10:1-4493-9041-2

Teaching Methodology	Classroom, seminar and assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103040003]

Course: 304: Advanced Database Administration

<table>
<tr><td>Course Code</td><td>304</td></tr>
<tr><td>Course Title</td><td>Advanced Database Administration</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks/ Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>Understanding advanced database administration</td></tr>
<tr><td>Course Objective</td><td>To learn advanced database administration, database tuning and maintenance</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Understand & Analyze): Explain the core concepts of Advanced Database Administration and analyze them from an industry-oriented perspective.
CO2 (Understand): Describe the internal architecture of the Oracle Database Management System, including its components and processes.
CO3 (Understand & Apply): Explain the concepts of database security and demonstrate user management techniques in Oracle.
CO4 (Apply): Utilize various Oracle utilities for database administration tasks and operations.
CO5 (Apply & Evaluate): Implement practical techniques for database tuning and evaluate performance optimization strategies.
CO6 (Apply & Create): Perform backup and recovery operations and develop recovery strategies for different failure scenarios.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>RDBMS</td></tr>
<tr><td>Course Content</td><td>Unit 1: Oracle10g Instance creation and management, Database Architecture, and Concurrency Management
1.1 Oracle 10g Instance management
1.1.1 Oracle Instance
1.1.2 Installing Oracle
1.1.3 Oracle Optimal Flexible Architecture (OFA)
1.1.4 Locating initialization, listener.ora & sqlnet.ora files
1.1.5 Finding the alert log
1.1.6 Common environment variables
1.1.7 Structures in an Oracle Instance
1.1.8 Oracle Memory Structures, SGA and PGA
1.1.9 Oracle Processes and their purposes
1.1.10 Startup, nomount, mount and open database commands
1.2 Oracle10g Database Architecture
1.2.1 Oraclde10g management framework
1.2.2 Using the Database Creation Assistant (DBA)
1.2.3 Creating and dropping a database
1.2.4 Tablespaces
1.2.5 Tables and Indexes
1.2.6 Clusters
1.2.7 Partitioning of Tables and Indexes
1.2.8 Gathering and applying patches
1.3 Concurrency Management
1.3.1 Transactions, serialization, locks and latches
1.3.2 Lock modes</td></tr>
</table>

1.3.3 Detecting and resolving lock conflicts
1.3.4 Managing deadlocks

Unit 2: Interfacing with Oracle, Oracle*Net and Tablespace Management

2.1 Transaction management and PL/SQL
2.1.1 Oracle transaction management
2.1.2 Using SQL*Plus and iSQL*Plus
2.1.3 Using embedded Oracle with Pro*C & Java
2.1.4 PL/SQL & Triggers
2.1.5 Pining PL/SQL packages & compiling PL/SQL
2.1.6 System-level triggers – startup trigger, logon trigger, PL/SQL error trigger

2.2 Oracle*Net
2.2.1 Basic Network structure
2.2.2 Oracle*Net Files
2.2.3 Multi-threaded server
2.2.4 Create additional listeners
2.2.5 Create Oracle Net service aliases
2.2.6 Configure connect time failover
2.2.7 Oracle*Net names resolution

2.3 Tablespace Management
2.3.1 Tablespace management
2.3.1.1 Dictionary Managed Tablespaces
2.3.1.2 Locally Managed Tablespaces
2.3.1.3 Automatic Segment Space Management
2.3.1.4 Moving tablespaces online and offline
2.3.2 UNDO tablespace management
2.3.2.1 Use of undo segments
2.3.2.2 Creating an undo tablespace
2.3.2.3 User managed undo tablespaces
2.3.2.4 Automatic undo management
2.3.2.5 Monitor & Configure undo retention
2.3.2.6 Use the Undo Advisor
2.3.2.7 Size the undo tablespace

Unit 3: Oracle Utilities, Performance Tuning

3.1 Utilities
3.1.1 Datapump - Import/export
3.1.2 SQL*Loader
3.1.3 Oracle Streams
3.1.4 Automatic Database Diagnostic Monitor
3.1.5 Automatic Tuning Optimizer
3.1.6 Automatic Shared Memory Tuning

3.2 Oracle Performance Tuning
3.2.1 Locate invalid and unusable objects
3.2.2 Gather SQL optimizer statistics with dbms_stats
3.2.3 Basic Oracle performance metrics
3.2.4 Use OEM and dbms_alert to set warning and critical alert thresholds
3.2.5 The SQL Tuning Advisor
3.2.6 The SQL Access Advisor
3.2.7 Interpreting server generated alerts
3.2.8 Oracle advisory utilities vdb_cache_advice, vshared pool_advice, v$pga_aggregate_target_advice
3.2.9 Using OEM performance screens
3.2.10 Fixing performance issues

	Unit 4: User Management, Security, Backup & Recovery 4.1 User management 4.1.1 Creating Users 4.1.2 Altering users 4.1.3 User Profiles 4.1.4 User resource groups 4.1.5 Granting privileges & roles 4.1.6 Auditing user activity with dbms_audit 4.2 Oracle Security 4.2.1 Password use in Oracle, Password encryption and password aging, External authentication, Using Single sign-on (SSO) 4.2.2 Object security 4.2.3 Virtual Private Databases (VPD) in Oracle 4.2.4 Oracle "grant execute" security 4.2.5 Use of Roles in Oracle 4.2.6 Register for security updates 4.3 Backup & Recovery 4.3.1 Oracle backup & recovery planning 4.3.2 Parallel instance recovery 4.3.3 Basics of checkpoints, redo log files, and archived log files 4.3.4 Using ARCHIVELOG mode 4.3.5 Creating consistent Oracle backups 4.3.6 Online hot backups 4.3.7 Incremental Oracle backups 4.3.8 Automating database backups with dbms_scheduler 4.3.9 Monitor the flash recovery area 4.3.10 Recovering from loss of a Control file 4.3.11 Recovering from loss of a Redo log file 4.3.12 Recovering from loss of a system-critical data file 4.3.13 Recovering from loss of a non system-critical data file
Reference Books	1. Essentials: Oracle Database 10g by Rick Greenwald, Robert Stackowiak, Jonathan Stern, O'Reilly 2. Oracle High Performance Tuning for 9i and 10g by Gavin Powell, Digital Press 3. Oracle Database 10g, DBA Handbook by Loney, Kevin, Bryla, Bob, Oracle Press 4. Oracle Database 10g - The Complete Reference by Loney, Kevin, Oracle Press 5. Oracle Database 10g: A Beginner's Guide by Micheal Abbey, Ian Abramson Osborne, Oracle Press Series
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103050003]

Course: 305: Data Warehousing & Data Mining

Course Code	305
Course Title	**Data Warehousing & Data Mining**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks/ Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Understanding Data Warehousing and Data Mining
Course Objective	To understand data warehousing and data mining - concepts and methods, and apply them in practice
Course Outcome	CO1 (Understand & Analyze): Differentiate between Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) systems, and describe multidimensional schemas suitable for data warehousing. CO2 (Understand & Apply): Explain data warehousing architectures and tools, and apply them to systematically organize large volumes of online data for strategic decision-making. CO3 (Understand & Apply): Illustrate various data pre-processing methods such as data reduction, cleaning, integration, and transformation, and apply them to prepare data for analysis. CO4 (Apply & Analyze): Utilize data mining techniques to extract meaningful knowledge and analyze patterns for prediction and classification tasks. CO5 (Apply & Create): Design and develop a data mining application for data analysis using appropriate algorithms and tools.
Mapping between COs with PSOs	
Pre-requisite	RDBMS, Basics of statistics
Course Content	**Unit 1: Data warehouse: Introduction** 1.1 Data Warehouse characteristics 1.2 Data Marts 1.3 OLTP and OLAP systems 1.4 Star, Snowflakes, and Fact Constellations Schemas for Multi-dimensional Databases 1.5 OLAP Operations in the Multidimensional data model 1.6 Type of OLAP servers **Unit 2: Developing Data Warehouse and Data Pre-processing** 2.1 Data Warehouse 2.1.1 Building a Data Warehouse 2.1.2 Three-Tier Data Warehouse Architecture 2.1.3 Metadata Repository 2.2 Data Pre-processing 2.2.1 Descriptive Data Summarization: central tendency, disperation of data 2.2.2 Data Cleaning: missing values, noisy data 2.2.3 Data Integration & Transformation 2.2.4 Data Reduction: Attribute selection 2.2.5 Data Discretization & Concept Hierarchy Generation

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■					■		■
CO2		■		■	■	■		■
CO3			■			■	■	■
CO4		■	■	■		■	■	■
CO5		■	■	■	■	■	■	■

	Unit 3: Data Mining, Associations, and Correlations 3.1 Knowledge discovery and Data Mining. 3.2 Basic Introduction to Data Mining Functionalities: 3.2.1 Concept/Class Description Characterization & Discrimination 3.2.2 Mining Frequent Patterns, Associations, and Correlations 3.2.3 Classification & Prediction 3.3.4 Cluster Analysis 3.2.5 Outlier Analysis 3.2.6 Evolution analysis 3.3 Mining Frequent Patterns, Associations, and Correlations 3.3.1 Basic concepts: Frequent Itemsets & Closed Itemsets, Association Rules 3.3.2 The Apriory algorithm: Finding Frequent Itemsets Using Candidate Generation 3.3.3 FP-growth: Finding Frequent Itemsets without Candidate Generation 3.3.4 Generating Association Rules from Frequent Itemsets 3.3.5 Introduction to multilevel and multidimensional Association rules **Unit 4: Classification & Prediction, Cluster Analysis and Recent Trends** 4.1 Classification and Prediction 4.1.1 Introduction to Classification & Prediction 4.1.2 Prediction: Linear Regression, Nonlinear Regression 4.1.3 Decision Tree Algorithm 4.1.3.1 Decision Tree Induction 4.1.3.2 Attribute Selection Measures- Information Gain and Gain Ratio 4.1.3.3 Tree Pruning 4.1.4 Bayesian Classification 4.1.4.1 Bayes' Theorem 4.1.4.2 Naïve Bayesian Classification 4.1.5 Accuracy and Error Measures for classification 4.2 Cluster Analysis 4.2.1 Classification vs. clustering 4.2.2 What is Partitioning & Hierarchical Clustering Methods 4.2.3 Classical Partitioning Methods: k-Means 4.3 Application and Trends in Data Mining
Reference Books	1. Data Mining: Concepts & Techniques by Han & Kamber, Morgan Kaufmann Publishers 2. Introduction to Data Mining with Case Studies by G. K. Gupta, PHI 3. Data Mining Introductory and Advanced Topics by Dunha, Pearson 4. Data Warehouse Toolkit by R. Kinball, John Wiley & Sons 5. Data Warehouses and OLAP: Concepts, Architectures, and Solutions by Robert Wrembel, Christian Koncilia, IGI 6. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management by Gordon S. Linoff, Michael J. A. Berry, Wiley 7. Data Preparation for Data Mining by Dorian Pyle, Morgan Kaufmann Publishers 8. Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals by Paulraj Ponniah, Wiley 9. Data Warehousing: Concepts, Techniques, Products and Applications by C.S.R. Prabhu, PHI Learning 10. Advanced Data Mining Techniques by David Louis Olson, Dursun Delen, Springer
Teaching Methodology	Class work, Discussion, Self-study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103050004]

Course: 305: Big Data

<table>
<tr><td>Course Code</td><td>305</td></tr>
<tr><td>Course Title</td><td>Big Data</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks/ Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>Understanding distributed computing, Big Data and Hadoop</td></tr>
<tr><td>Course Objective</td><td>To learn Big Bata and Hadoop</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Understand): Explain the fundamental concepts and building blocks of Big Data, including its characteristics and challenges.
CO2 (Apply): Demonstrate the programming aspects of cloud computing, with emphasis on MapReduce and other distributed processing models.
CO3 (Apply & Analyze): Access and process data on a Distributed File System, and analyze Big Data using various real-world applications.
CO4 (Apply & Evaluate): Represent and evaluate the analytical aspects of Big Data through data visualization and pattern discovery.
CO5 (Apply): Demonstrate an understanding of distributed computing principles and gain hands-on experience with the Hadoop ecosystem.
CO6 (Understand & Analyze): Describe recent trends in Big Data technologies, particularly Hadoop File System, MapReduce, and related frameworks, and analyze their impact on data processing.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>RDBMS</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 Evolution of Big Data
1.2 Structuring Big Data
1.3 Elements of Big Data(V’s)
1.4 Big Data Analytics
1.5 Commercial use of Big Data

Unit 2: Big Data Technology
2.1 Distributed and Parallel Computing
2.2 Introducing Hadoop
2.3 HDFS and MapReduce
2.4 Cloud Computing and Big Data
2.5 In-Memory Computing

Unit 3: Hadoop & Technology Foundations
3.1 HDFS Architecture
3.2 Blocks
3.3 Name Nodes and Data Nodes
3.4 Using HDFS Files
3.5 Hadoop Specific File System Types
3.6 HDFS Commands
3.7 org.apache.hadoop.io package
3.8 MapReduce Architecture
3.9 Hadoop YARN</td></tr>
</table>

	3.10 HBase 3.11 Combining HBase 3.12 Hive 3.13 Pig and Pig Latin 3.14 Sqoop 3.15 Big Data Stack 3.16 Virtualization and Big Data **Unit 4: Storing Data in Databases and Processing of Data** 4.1 RDBMS and Big Data 4.1.1 CAP Theorem 4.3 NoSQL Databases 4.4 Polygot Persistence 4.5 Integrating Big Data with traditional Data Warehouses 4.6 Big Data Analytics 4.7 Processing Data with MapReduce 4.8 Customizing MapReduce Execution and implementing MapReduce Program 4.9 Testing and Debugging MapReduce Applications 4.10 Analytical Approaches and Tools to Analyze Data
Reference Books	1. DT Editorial services, Big Data Black book, Dreamtech Press, ISBN 978-93-5119-931 2. Alex Holmes, Hadoop in Practice, Manning Publication company, 2014, ISBN 1617292222, 9781617292224 3. Kuan-Ching, Li Hai Jiang, Laurence T. Yang Alfredo Cuzzocrea, Big Data: Algorithms, Analytics and Applications. CRC Press 4. Hu, Wen Chen, Big Data Management, Technologies and Applications, IGI Global 5. Tom White, Hadoop The Definitive Guide, o'Reilly
Teaching Methodology	Class work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103060002]

Course: 306: Programming Skills VIII

Course Code	306
Course Title	**Programming Skills VIII**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn practically Embedded systems and IOT / ML
Course Objective	Learn Embedded systems and IOT practically, understand the working of Micro-Controller & Micro-Computer and using various types of sensors and its Programming. / Implement various algorithms used in Machine learning and introduce students with Programming in ML.
Pre-requisite	C/C++ Language / Python Programming
Course Outcome	After successful completion, students will be able to work with different types of Micro-Controllers, Micro-Computers and sensors for their IOT based application development. / After successful completion, students will be able to work with different types of ML algorithms and ML based application development.
Course Content	Practical based on paper no 301. (Embedded systems and IOT/ML) Separate journal to be prepared for this Course
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises , Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103070002]

Course: .307: Programming Skills IX

Course Code	307
Course Title	**Programming Skills IX**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to implement the concepts of SAP/NoSQL practically
Course Objective	Learn how to implement the ERP using SAP/NoSQL databases practically
Pre-requisite	DBMS
Course Outcome	After studying the course, students will be able to practically work on SAP/NoSQL Databases
Course Content	Practical based on paper no 303. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103080002]

Course: 308: Programming Skills X

Course Code	308
Course Title	**Programming Skills X**
Credit	2
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to handle advanced database administration activities
Course Objective	Advanced database administration
Pre-requisite	Basic database administration concepts
Course Outcome	After studying the course, students will be able to handle database administration, tuning and maintenance in various fields
Course Content	Practical based on paper no 304. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercise, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103090002]

Course: 309: Programming Skills XI

Course Code	309
Course Title	**Programming Skills XI**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to implement Data warehousing & data mining / Big Data practically.
Course Objective	Learn to use Data warehousing and data mining techniques in various practical environments / use Big data practically.
Pre-requisite	Databases, SQL, Advanced SQL
Course Outcome	After studying the course, students will be able to understand Data warehousing and data mining/Big Data practically
Course Content	Practical based on paper no 305. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

MCA 3rd Sem

(Network Group)

[Subject code-2111040103010002]

Course: 301: Embedded Systems and IOT

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Embedded Systems and IOT</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to embedded system and IOT. The course also gives students an idea about various components of microcontroller, microprocessor and IOT and explain its working. The course also explains the role of embedded systems in IOT ecosystem.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is to:
1. Understand the concept of embedded systems, including their internal architecture and programming of embedded processors.
2. Apply the knowledge of interfacing techniques to connect input/output devices with processors.
3. Understand the fundamentals of the Internet of Things (IOT), including its architecture and key components.
4. Analyze the architecture and operational features of microcontrollers and microcomputers.
5. Identify and classify various types of sensors used in embedded and IOT systems.
6. Develop and Implement basic IOT programs using suitable embedded platforms.</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Recall and explain the fundamentals of 8-bit microcontrollers, including their architecture, instruction sets, and interfacing with peripherals.
CO2 (Applying): Apply Embedded C programming techniques for memory interfacing, I/O devices, and serial communication in embedded systems.
CO3 (Understanding): Understand and describe the role of Real-Time Operating Systems (RTOS) in embedded systems, including task management and priority-based scheduling.
CO4 (Analyzing): Analyze the working principles and features of microcontroller platforms like Arduino and NodeMCU (ESP8266), and their use in embedded system design.
CO5 (Evaluating): Evaluate and select appropriate communication protocols (Wi-Fi, Bluetooth, ZigBee, GSM) for IoT systems based on application requirements.
CO6 (Creating): Design and implement IoT applications such as smart agriculture, home automation, and smart healthcare, using sensors, embedded systems, and communication protocols.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Computer Architecture, Programming in C, Programming with C++</td></tr>
<tr><td>Course Content</td><td>Unit 1: 8-Bit Embedded Processor
1.1 8-Bit Microcontroller
1.2 Architecture
1.3 Instruction Set and Programming
1.4 Programming Parallel Ports
1.5 Timers and Serial Port</td></tr>
</table>

	1.6 Interrupt Handling. **Unit 2: Embedded C Programming** 2.1 Memory And I/O Devices Interfacing 2.2 Programming Embedded Systems in C 2.3 Need For RTOS 2.4 Multiple Tasks and Processes 2.5 Context Switching 2.6 Priority Based Scheduling Policies. **Unit 3: IOT and Arduino Programming** 3.1 Introduction to Microcontrollers: Architecture, Features, and Programming Basics 3.2 Arduino: Development and Programming 3.3 Arduino IDE Setup 3.4 Arduino Architecture 3.5 Arduino Pin Diagram 3.6 Introduction to NodeMCU (ESP8266) 3.7 NodeMCU Specifications and Applications 3.8 NodeMCU ESP8266 Pinout **Unit 4: IOT Sensors, Communication and open platforms and Applications development** 4.1 Introduction to Various Sensors 4.2 Interfacing Sensors with Arduino and Node MCU 4.3 IOT Communication Models and APIs 4.4 IOT Communication Protocols – Bluetooth – WiFi – ZigBee–GPS – GSM modules 4.5 Introduction to raspberry Pi 4.6 Architecture of Raspberry Pi 4.7 Complete Design of Embedded Systems 4.8 Development of IOT Applications: Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare
Reference Books	1. Getting Started with Internet of Things – By Cuno Pfister, O'Reilly 2. Learning Internet of Things – By Peter Waher , Packt Publication 3. Internet of Things: A Hands-on Approach – By Arshdip Bahga and Vijay Madisetti 4. IOT Governance, Privacy and Security Issues, IERC 5. IOT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things, Cisco Press 6. Fundamentals of IOT Communication Technologies, Springer 7. Microcontrollers – Architecture, Programming, Interfacing and system design – By Raj Kamal , Pearson 8. Exploring C for Microcontrollers: A hands on approach, Springer 9. Arduino for Dummies, Wiley 10. Make: Getting Started With Arduino - The Open Source Electronics Prototyping Platform, Shroff/Maker Media 11. ESP8266: Get Started With ESP8266 Programming NodeMCU Using Arduino IDE, Createspace Independent Pub 12. Internet of Things Projects with ESP32, Packt Publishing Limited 13. Microprocessor Architecture, Programming and Applications with the 8085 - By Ramesh Gaonkar , Penram International Publishing 14. Raspberry Pi for Dummies , Wiley 15. Raspberry Pi User Guide – By Eben Upton and Garath Halfacree, Wiley
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103010003]

Course: 301: Machine Learning

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Machine Learning (ML)</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to ML. The course also gives students an idea about various methods and algorithms of Machine Learning and application development of ML.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is –
1. To make student understand ML
2. To understand the various Machine Learning method
3. To explain various algorithms used in Machine learning
4. To introduce students with Programming in ML</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Define Machine Learning and its types (Supervised, Unsupervised, and Semi-supervised) and identify the tools and applications associated with it.
CO2 (Applying): Apply data preprocessing techniques, select appropriate models, and train models using tools like Scikit-learn.
CO3 (Understanding): Explain the importance of data wrangling, including data cleaning, transformation, reduction, discretization, and feature selection, and utilize data visualization techniques.
CO4 (Analyzing): Analyze and differentiate between supervised learning algorithms, including regression (Simple, Multiple, Linear, Gradient Descent, Logistic) and classification algorithms (KNN, SVM, Decision Trees, Naïve Bayes).
CO5 (Applying): Implement clustering algorithms (K-means, Hierarchical) and association rule learning algorithms (FP-Growth, Apriori) for unsupervised learning tasks.
CO6 (Creating): Design and implement neural networks, including their architecture, feedforward networks, and backpropagation, for solving real-world problems.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Basics of Linear Algebra, Statistics and Mathematics, Python Programming</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 Definition of Machine Learning and types of Machine Learning: Supervised, Unsupervised and Semi-supervised
1.2 Applications and tools of Machine Learning (Scikit learn library)
1.3 Data Pre-processing, Selecting a model and training a model
1.4 Evaluating a performance of model and improving performance
1.5 Data Wrangling
1.5.1 Definition and goal of Data Wrangling
1.5.2 Importance of Data Wrangling
1.5.3 Data Pre-processing and Data Cleaning
1.5.3.1 Data Cleaning
1.5.3.2 Data Transformation
1.5.3.3 Data Reduction</td></tr>
</table>

	1.5.3.4 Data Discretization 1.5.3.5 Feature Selection 1.5.4 Data Visualization **Unit 2: Supervised Learning** 2.1 Supervised Learning: Classification and Regression 2.2 Regression 2.2.1 Simple and Multiple Regression 2.2.2 Linear Regression 2.2.3 Gradient Decent 2.2.4 Logistic Regression 2.3 Classification Algorithms 2.3.1 K-nearest Neighbor 2.3.2 Support Vector Machines 2.3.3 Decision Trees 2.3.4 Naïve Bayes Classifier 2.4 Introduction to Support Vector Machine **Unit 3: Unsupervised Learning** 3.1 Introduction to Unsupervised learning 3.2 Clustering 3.2.1 Selection of Clusters 3.2.2 Algorithms 3.2.2.1 K – means clustering 3.2.2.2 Hierarchical Clustering 3.3 Association Rule Learning 3.3.1 Algorithms 3.3.1.1 FP- Growth 3.3.1.2 Apriori Algorithm **Unit 4: Neural Network** 4.1 Introduction to Neural Network 4.2 Architecture of Neural Network 4.3 Feedforward network and Backpropagation with example 4.4 Applications of Neural Network
Reference Books	1. “Machine Learning” by Tom M. Mitchell, McGraw Hill 2. “Understanding Machine Learning” by Shai Shalev-Shwartz, Shai Ben-David 3. “Machine Learning” by Anuradha Srinivasaraghavan, Vincy Joseph 4. “Machine Learning using Python” by U Dinesh Kumar Manaranjan Pradhan 5. “Real-World Machine Learning” by Henrik Brink, Joseph Richards, Mark Fetherolf 6. “Python Machine Learning” by Sebastian Raschka and Vahid Mirjalili 7. “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems” by Aurelien Geron 8. “Machine Learning in Action” by Peter Harrington 9. “Introduction to Machine Learning with Python : A Guide for Data Scientists” by Andreas C. Muller, Sarah Guido
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103020002]

Course: 302: Design Patterns

<table>
<tr><td>Course Code</td><td>302</td></tr>
<tr><td>Course Title</td><td>Design Patterns</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>The purpose of the course is to make student understand how Patterns can be implemented in various object oriented programming languages to solve real world problems.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is -
1. To study various Design Patterns
2. How these Patterns can be used to design better systems through Object Oriented Programming Languages</td></tr>
<tr><td>Course Outcome</td><td>CO1: Understand the purpose and use of design patterns and remember the SOLID patterns which are the building block for design patterns. Remember the various design patterns; their categories, and standard patterns as per GOF.
CO2: Analyze the first category of creational design patterns and learn them.
CO3: Understand the structural design patterns.
CO4: Understand and analyze the behavioral design patterns.
CO5: Learn some more design patterns used in IT industry currently.
CO6: Make students understand the applicability of design patterns practiced by IT companies and how effectively combine these patterns for effective software development.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Object Oriented Programming, Software Engineering</td></tr>
<tr><td>Course Content</td><td>Unit -1: Introduction to Design Patterns
1.1 What’s a Design Pattern?
1.2 Benefits of Design Patterns
1.3 Software Design Principles
1.4 SOLID Principles
1.5 Design patterns – their categories and standard design patterns as per GOF

Unit-2: Creational and Structural Patterns
2.1 Creational Patterns
2.1.1 Factory
2.1.2 Abstract Factory
2.1.3 Builder
2.1.4 Prototype
2.1.5 Singleton
2.2 Structural Patterns
2.2.1 Adapter
2.2.2 Bridge
2.2.3 Composite
2.2.4 Decorator
2.2.5 Overview of other Structural Patterns - Façade, Flyweight and Proxy patterns

Unit-3: Behavioral Patterns</td></tr>
</table>

	3.1 Chain of Responsibility 3.2 Command 3.3 Iterator 3.4 Mediator 3.5 Overview of other Behavioral Patterns - Memento, Observer, State, Strategy, Template Method and Visitor Patterns **Unit 4: Additional Patterns and Patterns Applicability** 4.1 Overview of Additional Patterns - Simple Factory, Null Object and MVC Patterns 4.2 Security Patterns Repository 4.3 Patterns for Agile Development 4.4 Relation between patterns and pattern combinations Self-Study : Implementation of Patterns in various Programming Languages
Reference Books	1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph, John, Addision Wesley 2. Drive into Design Patterns, Alexander Shvets, Refactoring.Guru, 2019 3. Head First Design Patterns, Eric Freeman, O'Reilly 4. Design Patterns in C#, Vaskaran Sarcar, Apress 5. Design Patterns in Modern C++, Reusable Approaches for Object-Oriented Software Design, Dmitri Nesteruk, Apress 6. Modern C++ design: generic Programming and design patterns applied, Alexendrescu, Andrei, Addison-Wesley 7. Java Design Patterns: A Hands-on Experience with Real-World Examples, Vaskaran Sarcar, Apress
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103030006]

Course: 303: **Network Essentials and its Security**

Course Code	303
Course Title	**Network Essentials and its Security**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	To make students learn Network essentials and various security measures for the challenges to which the IT industry is exposed
Course Objective	To understand Network Management and its security
Course Outcomes	CO1: *(Remembering/Understanding)* Describe the functions and operations of networking devices such as repeaters, bridges, routers, and gateways. CO2: *(Understanding)* Explain network management techniques including SNMP, backup systems, fault-tolerant mechanisms, and RAID configurations. CO3: *(Applying)* Demonstrate the working of various cryptographic techniques such as DES, RSA, and digital signatures to ensure message integrity and confidentiality. CO4: *(Analyzing)* Analyze various security protocols and mechanisms like IPsec, SSL/TLS, VPNs, and firewalls used in securing computer networks and internet communication. CO5: *(Evaluating)* Evaluate the effectiveness of authentication services and network security models including Kerberos, X.509, and SNMP security. CO6: *(Creating)* Design secure network environments considering mobile systems, intrusion detection, virus protection, and e-commerce transaction security.
Mapping between COs with PSOs	
Pre-requisite	Computer Network, TCP/IP
Course Content	**Unit 1: Network Essentials** 1.1 Repeaters & Bridges 1.1.1 LAN Expansion 1.1.2 Repeaters 1.1.3 Bridges 1.1.4 How Bridges Work 1.1.5 Creating the routing table 1.1.6 Segmenting Network Traffic 1.1.7 Remote Bridges 1.1.8 Differentiating between bridges and repeaters 1.2 Routers & Gateways 1.2.1 Routers 1.2.2 How routers work 1.2.3 Routing benefits 1.2.4 Routing protocols 1.2.5 Routing V/S Bridging

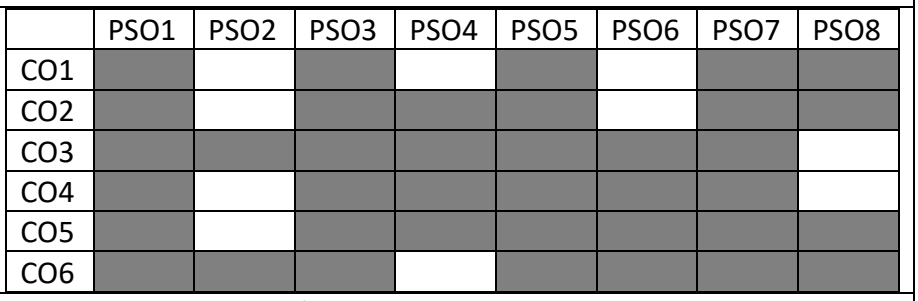
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■	■		■	■	■	■	
CO3	■	■	■			■	■	■
CO4	■	■	■	■		■	■	■
CO5	■	■	■	■	■			■
CO6							■	■

	1.2.6 B Routers 1.2.7 Gateways 1.2.8 How Gateways work 1.3 Network Administration 1.3.1 Bottlenecks 1.3.2 Simple Network Management Protocol 1.3.3 Data Protection 1.3.4 Backup Methods 1.3.5 Testing and Storage 1.3.6 Implementing a Backup System 1.3.7 Uninterruptible Power Suppliers 1.3.8 How Gateways work 1.3.9 Implementing Fault Tolerant Systems 1.3.10 RAID 1.3.11 Sector Sparing 1.4 Advance WAN Transmission 1.4.1 Overview 1.4.2 Multiplexing, Packet and Circuit Switching Networks 1.4.3 X.25 1.4.4 Asynchronous Transfer Mode (ATM) 1.4.5 ISDN 1.4.6 SONET 1.4.7 SMDS 1.5 Introduction to Network Security **Unit 2: Cryptography Techniques and Security Services** 2.1 Classical Cryptography 2.2 Conventional Cryptography 3.2.1 DES 2.3 Public – key Cryptography 3.3.1 RSA 2.4 Digital Signatures 3.4.1 DSA 2.5 Message Integrity 2.6 Confidentiality and Authentication 2.7 Certification and Key Management **Unit 3: Network Security Applications & Access Control in Computer Networks** 3.1 IP Security 3.1.1 IPsec 3.2 Web Security 3.2.1 SSL, TLS, SET 3.3 Electronic Mail Security 3.3.1 PGP, S/MIME 3.4 SNMP Security 3.5 Authentication Protocols and Services 3.5.1 Kerberos and X.509 3.6 Firewalls 3.7 Virtual Private Networks (VPNs) **Unit 4: Mobile System & E-Commerce Securities & System Security** 4.1 Intrusion detection 4.2 Viruses 4.1 3G Security 4.2 E-Payment Systems 4.3 Fair Data Exchange

Reference Books	1. Cryptography and Network Security, 2/e, ISBN: 0-13-869017-0 - W. Stallings - Pearson Education, 1999 2. Network Security Essentials: Applications and Standards, 1/e, ISBN: 0-13-016093-8 - W. Stallings - Pearson Education, 2000 3. SSL and TLS: designing and building secure systems, ISBN: 0-201-61598-3 - E. Rescorla - Addison-Wesley, 2001 4. Implementing Secure Intranets and Extranets, ISBN: 0-89006-447-4 – 5. K M Phaltankar - Artech House Publishers, 2000 6. Secure Electronic Commerce: Building the Infrastructure for Digital 7. Signature and Encryption, ISBN: 0-13-027276-0 - W. Ford, and M. Baum - Prentice Hall, 2001 8. Security in Computing, ISBN: 0-13-185794-0, 2/e - C. P. Pfleeger - Prentice Hall, 1997 9. Building Internet Firewalls, 2/e, ISBN: 1-56592-871-7 - E. D. Zwicky, et al - O'Reilly, 2000 10. CDMA Cellular Mobile Communications & Network Security, ISBN: 0-13-598418-1 - M. Y. Rhee, - Prentice Hall, 1998 11. Journal of Computer Security 12. ACM Transactions on Information and System Security 13. ACM Conference on Computer and Communications Security 14. IEEE Symposium on Security and Privacy 15. Internet documents - RFCs (Request for Comments) 16. Guide to Networking Essentials, Fourth Edition - Greg Tomsho, et al 17. Computer Networking Essentials - Debra Littlejohn Shinder 18. Networking Essentials: Hands-On, Self-Paced Training for 19. Supporting Local and Wide Area Networks - Microsoft Corporation (Corporate Author) 20. Computer Network - A. S. Tanenbaum
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103040004]

Course: 304: **Network Administration**

Course Code	304
Course Title	**Network Administration**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Understanding advanced network administration
Course Objective	To learn advanced network administration, its configuration and maintenance
Course Outcomes	CO1: (*Remembering / Understanding*) Recall and explain the fundamentals of Linux networking, TCP/IP architecture, and dial-up technologies. CO2: (*Applying*) Configure essential network services like DHCP and DNS on Linux systems. CO3: (*Applying)* Apply Linux remote administration tools such as Telnet, SSH, and services like NFS and NIS for managing networked systems. CO4: (*Analyzing*) Analyze the role of cron daemon, system-wide logging, and network automation in maintaining and securing Linux environments. CO5: (*Evaluating)* Evaluate the effectiveness of Samba and cross-platform file sharing methods in heterogeneous networks. CO6: (*Creating)*Design a secure and efficient Linux-based network setup integrating services like DNS, NFS, Samba, cron, and remote administration tools.
Mapping between COs with PSOs	
Pre-requisite	Computer Network, TCP/IP
Course Content	**Unit 1: Networking and TCP/IP on Linux** 1.1 Fundamentals of Linux Networking 1.2 Fundamentals of TCP/IP on the Linux Operating System 1.3 Advanced Linux TCP/IP Concepts 1.4 Introduction to Dial-up Technologies **Unit 2: Dynamic Host Configuration Protocol & Domain Name System** 2.1 Introduction to BOOTP and DHCP 2.2 Installing and Examining a Linux DHCP server 2.3 Examining Additional DHCP Options and Configurations 2.4 Introduction to the Domain Name System 2.5 Installing and Configuring DNS **Unit 3: Network Services and Remote Administration in Linux** 3.1 The Network File System (NFS) 3.1.1 Introduction to the Network File System 3.1.2 Configuring NFS 3.2 Linux Remote Administration 3.2.1 Introduction to Remote Administration 3.2.2 The Telnet Protocol 3.2.3 The open Secure Shell (SSH) Protocol

	3.3 The Cron Daemon 3.3.1 Introduction to Automation 3.3.2 Configuring the Cron Daemon 3.4 The Network Information Service (NIS) 3.4.1 Introduction to NIS 3.4.2 Setting Up and Configuring an NIS Server **Unit 4: File and Resource Sharing with Logging in Linux** 4.1 Samba 4.1.1 Introduction to Samba 4.1.2 Cross-Platform Connectivity 4.1.3 Installing and Configuring Samba 4.2 Linux System-Wide Logging 4.2.1 Introduction to System-wide Logging 4.2.2 Configuring System Logging 4.3 The Network Information Service 4.3.1 Introduction to NIS 4.3.2 Setting Up and Configuring an NIS server
Reference Books	1. TCP/IP Network Administration - Craig Hunt - O'Reilly & Associates 2. Managing NFS and NIS - Hal Stern - O'Reilly & Associates 3. DNS and BIND - Albitz/Liu - O'Reilly & Associates 4. Sendmail - Bryan Costales/Eric , Allman/Neil Rickert - O'Reilly & Associates 5. UNIX System Administration Handbook - Second Edition - Nemeth/Snyder/Seebass - Prentice Hall 6. Red Hat Linux Networking and System Administration – Terry Collings, Kurt Wall 7. Red Hat Linux 9 Bible - Christopher Negus 8. Official Red Hat Linux User's Guide - Red Hat Inc. 9. Official Red Hat Linux Administrator's Guide - Red Hat Inc. 10. Red Hat Linux Security and Optimization - Mohammad J. Kabir
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103050005]

Course: 305: **Wireless Network and Mobile Computing**

Course Code	305
Course Title	**Wireless Network and Mobile Computing**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Understanding Wireless Network and Mobile Computing
Course Objective	To understand various aspects related to Wireless Network technologies, mobile networks and its computing
Course Outcomes	CO1: (*Remembering / Understanding*) Identify and describe wireless networking technologies, standards, and protocols including OSI model, LAN, and IP. CO2: (*Understanding*) Explain the operation of Wireless LAN technologies such as DSSS, FHSS, Mobile IP, modulation, and spread spectrum methods. CO3: (*Applying)* Configure and implement wireless network hardware like access points, bridges, antennas, and gateways. CO4: (*Analyzing*)Analyze MAC and routing protocols (e.g., DSDV, DSR, AODV, TORA) in the context of IEEE 802.11 and MANETs. CO5: (*Evaluating*) Evaluate mobile computing architectures, protocols, and standards used in GSM, GPRS, and WAP systems. CO6: *(Creating)* Design a basic wireless or mobile computing network integrating protocols, hardware, and mobile internet standards.
Mapping between COs with PSOs	
Pre-requisite	Computer Network
Course Content	**Unit 1: Introduction to Wireless Network Technologies** 1.1 Introduction 1.2 Standards 1.3 Emerging Technologies 1.4 OSI Basics 1.5 LAN Basics 1.6 LAN & WAN Protocols 1.7 Internet Protocol **Unit 2: Wireless LAN Technologies, Implementation and Layers** 2.1 Frequency Hopping Spread Spectrum 2.2 Direct Sequence Spread Spectrum (DSSS) 2.3 Interference 2.4 RF Math 2.5 Service Sets 2.6 Mobile IP 2.7 Appropriate use and design of wireless NetworkingMultipath 2.8 Co-location 2.9 Power-over-Ethernet (PoE) 2.10 Modulation and Bit Coding 2.11 Fragmentation 2.12 SIFS / PIFS / DIFS / EIFS

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■						■	
CO2	■	■	■	■		■	■	
CO3	■	■	■	■	■	■	■	
CO4	■	■		■	■	■	■	■
CO5	■	■	■	■	■	■	■	■
CO6	■	■	■	■	■	■	■	■

	Unit 3: Hardware Configuration, Implementation and Protocols & MANET 3.1 Access Points 3.2 Bridges 3.3 Workgroup bridges 3.4 Wireless Residential Gateways 3.5 Host Connectivity 3.6 Antennas, Cables, & Connectors 3.8 MAC and Routing Protocols for IEEE 802.11 3.8 Wireless Mesh Networks 3.9 Various applications of MANET 3.10 Destination- Sequenced Distance Vector protocol 3.11 Dynamic Source Routing protocol 3.12 Ad Hoc On-Demand Distance-Vector protocol 3.13 Link Reversal Routing 3.13.1 Gafni-Bertsekas algorithm 3.13.2 Lightweight mobile routing algorithm. 3.13.3 Temporally ordered routing algorithm **Unit 4: Introduction to Mobile Computing, GSM, GPRS & WAP** 4.1 Mobility, Nomadic, Mobile and Ubiquitous computing 4.2 Mobile Computing Architecture 4.3 Mobile Computing Technologies (Hardware, Software, Communication) 4.4 Introduction to GSM 4.5 GSM Architecture, Mobility Management, Network Signaling 4.6 GPRS Architecture 4.7 Network Nodes 4.8 Mobile Internet Standards, WAP Gateway and Protocols 4.9 WML
Reference Books	1. Wireless Local Area Network Fundamentals - Pejman Roshan, Jonathan Leary 2. Wireless Networks First Step (First-step series) - Jim Geier 3. 802.11 wireless network site surveying and installation - Bruce Alexander 4. Introduction to Wireless and Mobile Systems by Cengage Learning (Thompson) 5. J. Schiller, Mobile Communications, Addison –Wesley, 2003 6. Wi-Fi Security - Stewart Miller 7. Wireless and Mobile Network - Architecture Yi-Bing Lin & Imrich Chlamtac - John Wiley & Sons, 2001 8. Mobile and Wireless Design Essentials by Martyn Mallick, John Wiley & Sons 9. Guide to Designing and Implementing wireless LANs - Mark Ciampa - Thomson learning , Vikas Publishing House, 2001 10. Wireless Web Development - Ray Rischapter - Springer publishing, 2000
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103060002]

Course: 306: Programming Skills VIII

Course Code	306
Course Title	**Programming Skills VIII**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn practically Embedded systems and IOT / ML
Course Objective	Learn Embedded systems and IOT practically, understand the working of Micro-Controller & Micro-Computer and using various types of sensors and its Programming. / Implement various algorithms used in Machine learning and introduce students with Programming in ML.
Pre-requisite	C/C++ Language / Python Programming
Course Outcome	After successful completion, students will be able to work with different types of Micro-Controllers, Micro-Computers and sensors for their IOT based application development. / After successful completion, students will be able to work with different types of ML algorithms and ML based application development.
Course Content	Practical based on paper no 301. (Embedded systems and IOT/ML) Separate journal to be prepared for this Course
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103070002]

Course: 307: **Programming Skills IX**

Course Code	307
Course Title	**Programming Skills IX**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to understand the Networking and its security practically
Course Objective	Learning to implement Network and its security practically.
Pre-requisite	Network fundamentals
Course Outcome	After studying the course, students will be able to practically implement network and its security
Course Content	Practical based on paper no 301. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Exercises , Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103080002]

Course: 308: **Programming Skills X**

Course Code	308
Course Title	**Programming Skills X**
Credit	2
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to learn network administration
Course Objective	Students will learn administration of network practically
Pre-requisite	Networking fundamentals
Course Outcome	After studying the course, students will be able to practically perform administrative tasks of networks practically
Course Content	Practical based on paper no 304. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103090002]

Course: 309: **Programming Skills XI**

Course Code	309
Course Title	**Programming Skills XI**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to implement the basic and advanced concepts of wireless network and mobile computing
Course Objective	Learn how to impliment wireless protocols and its implementation practically
Pre-requisite	Computer Network, C/C++ programming
Course Outcome	After studying the course, students will be able to practically develop/enhance wireless protocols and find better solutions application to the various industries dependent upon them
Course Content	Practical based on paper no 305. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Exercises, Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

MCA 3rd Sem. (General Group)

[Subject code-2111040103010002]

Course: 301: Embedded Systems and IOT

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Embedded Systems and IOT</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to embedded system and IOT. The course also gives students an idea about various components of microcontroller, microprocessor and IOT and explain its working. The course also explains the role of embedded systems in IOT ecosystem.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is to:
1. Understand the concept of embedded systems, including their internal architecture and programming of embedded processors.
2. Apply the knowledge of interfacing techniques to connect input/output devices with processors.
3. Understand the fundamentals of the Internet of Things (IOT), including its architecture and key components.
4. Analyze the architecture and operational features of microcontrollers and microcomputers.
5. Identify and classify various types of sensors used in embedded and IOT systems.
6. Develop and Implement basic IOT programs using suitable embedded platforms.</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Recall and explain the fundamentals of 8-bit microcontrollers, including their architecture, instruction sets, and interfacing with peripherals.
CO2 (Applying): Apply Embedded C programming techniques for memory interfacing, I/O devices, and serial communication in embedded systems.
CO3 (Understanding): Understand and describe the role of Real-Time Operating Systems (RTOS) in embedded systems, including task management and priority-based scheduling.
CO4 (Analyzing): Analyze the working principles and features of microcontroller platforms like Arduino and NodeMCU (ESP8266), and their use in embedded system design.
CO5 (Evaluating): Evaluate and select appropriate communication protocols (Wi-Fi, Bluetooth, ZigBee, GSM) for IoT systems based on application requirements.
CO6 (Creating): Design and implement IoT applications such as smart agriculture, home automation, and smart healthcare, using sensors, embedded systems, and communication protocols.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Computer Architecture, Programming in C, Programming with C++</td></tr>
<tr><td>Course Content</td><td>Unit 1: 8-Bit Embedded Processor
1.1 8-Bit Microcontroller
1.2 Architecture
1.3 Instruction Set and Programming
1.4 Programming Parallel Ports
1.5 Timers and Serial Port</td></tr>
</table>

	1.6 Interrupt Handling. **Unit 2: Embedded C Programming** 2.1 Memory And I/O Devices Interfacing 2.2 Programming Embedded Systems in C 2.3 Need For RTOS 2.4 Multiple Tasks and Processes 2.5 Context Switching 2.6 Priority Based Scheduling Policies. **Unit 3: IOT and Arduino Programming** 3.1 Introduction to Microcontrollers: Architecture, Features, and Programming Basics 3.2 Arduino: Development and Programming 3.3 Arduino IDE Setup 3.4 Arduino Architecture 3.5 Arduino Pin Diagram 3.6 Introduction to NodeMCU (ESP8266) 3.7 NodeMCU Specifications and Applications 3.8 NodeMCU ESP8266 Pinout **Unit 4: IOT Sensors, Communication and open platforms and Applications development** 4.1 Introduction to Various Sensors 4.2 Interfacing Sensors with Arduino and Node MCU 4.3 IOT Communication Models and APIs 4.4 IOT Communication Protocols – Bluetooth – WiFi – ZigBee– GPS – GSM modules 4.5 Introduction to raspberry Pi 4.6 Architecture of Raspberry Pi 4.7 Complete Design of Embedded Systems 4.8 Development of IOT Applications: Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare
Reference Books	1. Getting Started with Internet of Things – By Cuno Pfister, O'Reilly 2. Learning Internet of Things – By Peter Waher , Packt Publication 3. Internet of Things: A Hands-on Approach – By Arshdip Bahga and Vijay Madisetti 4. IOT Governance, Privacy and Security Issues, IERC 5. IOT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things, Cisco Press 6. Fundamentals of IOT Communication Technologies, Springer 7. Microcontrollers – Architecture, Programming, Interfacing and system design – By Raj Kamal , Pearson 8. Exploring C for Microcontrollers: A hands on approach, Springer 9. Arduino for Dummies, Wiley 10. Make: Getting Started With Arduino - The Open Source Electronics Prototyping Platform, Shroff/Maker Media 11. ESP8266: Get Started With ESP8266 Programming NodeMCU Using Arduino IDE, Createspace Independent Pub 12. Internet of Things Projects with ESP32, Packt Publishing Limited 13. Microprocessor Architecture, Programming and Applications with the 8085 - By Ramesh Gaonkar , Penram International Publishing 14. Raspberry Pi for Dummies , Wiley 15. Raspberry Pi User Guide – By Eben Upton and Garath Halfacree, Wiley
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment

Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103010003]

Course: 301: Machine Learning

(Elective)

<table>
<tr><td>Course Code</td><td>301</td></tr>
<tr><td>Course Title</td><td>Machine Learning (ML)</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>This course is an introduction for students to ML. The course also gives students an idea about various methods and algorithms of Machine Learning and application development of ML.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is –
1. To make student understand ML
2. To understand the various Machine Learning method
3. To explain various algorithms used in Machine learning
4. To introduce students with Programming in ML</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Remembering / Understanding): Define Machine Learning and its types (Supervised, Unsupervised, and Semi-supervised) and identify the tools and applications associated with it.
CO2 (Applying): Apply data preprocessing techniques, select appropriate models, and train models using tools like Scikit-learn.
CO3 (Understanding): Explain the importance of data wrangling, including data cleaning, transformation, reduction, discretization, and feature selection, and utilize data visualization techniques.
CO4 (Analyzing): Analyze and differentiate between supervised learning algorithms, including regression (Simple, Multiple, Linear, Gradient Descent, Logistic) and classification algorithms (KNN, SVM, Decision Trees, Naïve Bayes).
CO5 (Applying): Implement clustering algorithms (K-means, Hierarchical) and association rule learning algorithms (FP-Growth, Apriori) for unsupervised learning tasks.
CO6 (Creating): Design and implement neural networks, including their architecture, feedforward networks, and backpropagation, for solving real-world problems.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Basics of Linear Algebra, Statistics and Mathematics, Python Programming</td></tr>
<tr><td>Course Content</td><td>Unit 1: Introduction
1.1 Definition of Machine Learning and types of Machine Learning: Supervised, Unsupervised and Semi-supervised
1.2 Applications and tools of Machine Learning (Scikit learn library)
1.3 Data Pre-processing, Selecting a model and training a model
1.4 Evaluating a performance of model and improving performance
1.5 Data Wrangling</td></tr>
</table>

	1.5.1 Definition and goal of Data Wrangling 1.5.2 Importance of Data Wrangling 1.5.3 Data Pre-processing and Data Cleaning 1.5.3.1 Data Cleaning 1.5.3.2 Data Transformation 1.5.3.3 Data Reduction 1.5.3.4 Data Discretization 1.5.3.5 Feature Selection 1.5.4 Data Visualization **Unit 2: Supervised Learning** 2.1 Supervised Learning: Classification and Regression 2.2 Regression 2.2.1 Simple and Multiple Regression 2.2.2 Linear Regression 2.2.3 Gradient Decent 2.2.4 Logistic Regression 2.3 Classification Algorithms : 2.3.1 K-nearest Neighbor 2.3.2 Support Vector Machines 2.3.3 Decision Trees 2.3.4 Naïve Bayes Classifier 2.4 Introduction to Support Vector Machine **Unit 3: Unsupervised Learning** 3.1 Introduction to Unsupervised learning 3.2 Clustering 3.2.1 Selection of Clusters 3.2.2 Algorithms: 3.2.2.1 K – means clustering 3.2.2.2 Hierarchical Clustering 3.3 Association Rule Learning 3.3.1 Algorithms : 3.3.1.1 FP- Growth 3.3.1.2 Apriori Algorithm **Unit 4: Neural Network** 4.1 Introduction to Neural Network 4.2 Architecture of Neural Network 4.3 Feedforward network and Backpropagation with example 4.4 Applications of Neural Network
Reference Books	1. “Machine Learning” by Tom M. Mitchell, McGraw Hill 2. “Understanding Machine Learning” by Shai Shalev-Shwartz, Shai Ben-David 3. “Machine Learning” by Anuradha Srinivasaraghavan, Vincy Joseph 4. “Machine Learning using Python” by U Dinesh Kumar Manaranjan Pradhan 5. “Real-World Machine Learning” by Henrik Brink, Joseph Richards, Mark Fetherolf 6. “Python Machine Learning” by Sebastian Raschka and Vahid Mirjalili 7. “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems” by Aurelien Geron 8. “Machine Learning in Action” by Peter Harrington 9. “Introduction to Machine Learning with Python : A Guide for Data Scientists” by Andreas C. Muller, Sarah Guido
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment

Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103020002]

Course: 302: Design Patterns

<table>
<tr><td>Course Code</td><td>302</td></tr>
<tr><td>Course Title</td><td>Design Patterns</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>The purpose of the course is to make student understand how Patterns can be implemented in various object oriented programming languages to solve real world problems.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is -
1. To study various Design Patterns
2. How these Patterns can be used to design better systems through Object Oriented Programming Languages</td></tr>
<tr><td>Course Outcome</td><td>CO1: Understand the purpose and use of design patterns and remember the SOLID patterns which are the building block for design patterns. Remember the various design patterns; their categories, and standard patterns as per GOF.
CO2: Analyze the first category of creational design patterns and learn them.
CO3: Understand the structural design patterns.
CO4: Understand and analyze the behavioral design patterns.
CO5: Learn some more design patterns used in IT industry currently.
CO6: Make students understand the applicability of design patterns practiced by IT companies and how effectively combine these patterns for effective software development.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Object Oriented Programming, Software Engineering</td></tr>
<tr><td>Course Content</td><td>Unit -1: Introduction to Design Patterns
1.1 What’s a Design Pattern?
1.2 Benefits of Design Patterns
1.3 Software Design Principles
1.4 SOLID Principles
1.5 Design patterns – their categories and standard design patterns as per GOF

Unit-2: Creational and Structural Patterns
2.1 Creational Patterns
2.1.1 Factory
2.1.2 Abstract Factory
2.1.3 Builder
2.1.4 Prototype
2.1.5 Singleton
2.2 Structural Patterns
2.2.1 Adapter
2.2.2 Bridge
2.2.3 Composite
2.2.4 Decorator</td></tr>
</table>

	2.2.5 Overview of other Structural Patterns - Façade, Flyweight and Proxy patterns **Unit-3: Behavioral Patterns** 3.1 Chain of Responsibility 3.2 Command 3.3 Iterator 3.4 Mediator 3.5 Overview of other Behavioral Patterns - Memento, Observer, State, Strategy, Template Method and Visitor Patterns **Unit 4: Additional Patterns and Patterns Applicability** 4.1 Overview of Additional Patterns - Simple Factory, Null Object and MVC Patterns 4.2 Security Patterns Repository 4.3 Patterns for Agile Development 4.4 Relation between patterns and pattern combinations Self-Study : Implementation of Patterns in various Programming Languages
Reference Books	1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph, John, Addision Wesley 2. Drive into Design Patterns, Alexander Shvets, Refactoring.Guru, 2019 3. Head First Design Patterns, Eric Freeman, O'Reilly 4. Design Patterns in C#, Vaskaran Sarcar, Apress 5. Design Patterns in Modern C++, Reusable Approaches for Object-Oriented Software Design, Dmitri Nesteruk, Apress 6. Modern C++ design: generic Programming and design patterns applied, Alexendrescu, Andrei, Addison-Wesley 7. Java Design Patterns: A Hands-on Experience with Real-World Examples, Vaskaran Sarcar, Apress
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103030006]

Course: 303: **Network Essentials and its Security**

Course Code	303
Course Title	**Network Essentials and its Security**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	To make students learn Network essentials and various security measures for the challenges to which the IT industry is exposed
Course Objective	To understand Network Management and its security
Course Outcomes	CO1: *(Remembering/Understanding)* Describe the functions and operations of networking devices such as repeaters, bridges, routers, and gateways. CO2: *(Understanding)* Explain network management techniques including SNMP, backup systems, fault-tolerant mechanisms, and RAID configurations. CO3: *(Applying)* Demonstrate the working of various cryptographic techniques such as DES, RSA, and digital signatures to ensure message integrity and confidentiality. CO4: *(Analysing)* Analyze various security protocols and mechanisms like IPsec, SSL/TLS, VPNs, and firewalls used in securing computer networks and internet communication. CO5: *(Evaluating)* Evaluate the effectiveness of authentication services and network security models including Kerberos, X.509, and SNMP security. CO6: *(Creating)* Design secure network environments considering mobile systems, intrusion detection, virus protection, and e-commerce transaction security.
Mapping between COs with PSOs	
Pre-requisite	Computer Network, TCP/IP
Course Content	**Unit 1: Network Essentials** 1.1 Repeaters & Bridges 1.1.1 LAN Expansion 1.1.2 Repeaters 1.1.3 Bridges 1.1.4 How Bridges Work 1.1.5 Creating the routing table 1.1.6 Segmenting Network Traffic 1.1.7 Remote Bridges 1.1.8 Differentiating between bridges and repeaters 1.2 Routers & Gateways 1.2.1 Routers 1.2.2 How routers work 1.2.3 Routing benefits 1.2.4 Routing protocols 1.2.5 Routing V/S Bridging

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■	■		■	■	■	■	
CO3	■	■	■			■	■	■
CO4	■	■	■	■		■	■	■
CO5	■	■	■	■	■			■
CO6							■	■

	1.2.6 B Routers 1.2.7 Gateways 1.2.8 How Gateways work 1.3 Network Administration 1.3.1 Bottlenecks 1.3.2 Simple Network Management Protocol 1.3.3 Data Protection 1.3.4 Backup Methods 1.3.5 Testing and Storage 1.3.6 Implementing a Backup System 1.3.7 Uninterruptible Power Suppliers 1.3.8 How Gateways work 1.3.9 Implementing Fault Tolerant Systems 1.3.10 RAID 1.3.11 Sector Sparing 1.4 Advance WAN Transmission 1.4.1 Overview 1.4.2 Multiplexing, Packet and Circuit Switching Networks 1.4.3 X.25 1.4.4 Asynchronous Transfer Mode (ATM) 1.4.5 ISDN 1.4.6 SONET 1.4.7 SMDS 1.5 Introduction to Network Security **Unit 2: Cryptography Techniques and Security Services** 2.1 Classical Cryptography 2.2 Conventional Cryptography 3.2.1 DES 2.3 Public – key Cryptography 3.3.1 RSA 2.4 Digital Signatures 3.4.2 DSA 1.5 Message Integrity 1.6 Confidentiality and Authentication 2.7 Certification and Key Management **Unit 3: Network Security Applications & Access Control in Computer Networks** 3.1 IP Security 3.1.1 IPsec 3.2 Web Security 3.2.1 SSL, TLS, SET 3.3 Electronic Mail Security 3.3.1 PGP, S/MIME 3.4 SNMP Security 3.5 Authentication Protocols and Services 3.5.1 Kerberos and X.509 3.6 Firewalls 3.7 Virtual Private Networks (VPNs) **Unit 4: Mobile System & E-Commerce Securities & System Security** 4.1 Intrusion detection 4.2 Viruses 4.1 3G Security 4.2 E-Payment Systems 4.3 Fair Data Exchange

Reference Books	1. Cryptography and Network Security, 2/e, ISBN: 0-13-869017-0 - W. Stallings - Pearson Education, 1999 2. Network Security Essentials: Applications and Standards, 1/e, ISBN: 0-13-016093-8 - W. Stallings - Pearson Education, 2000 3. SSL and TLS: designing and building secure systems, ISBN: 0-201-61598-3 - E. Rescorla - Addison-Wesley, 2001 4. Implementing Secure Intranets and Extranets, ISBN: 0-89006-447-4 - K M Phaltankar - Artech House Publishers, 2000 5. Secure Electronic Commerce: Building the Infrastructure for Digital Signature and Encryption, ISBN: 0-13-027276-0 - W. Ford, and M. Baum - Prentice Hall, 2001 6. Security in Computing, ISBN: 0-13-185794-0, 2/e - C. P. Pfleeger - Prentice Hall, 1997 7. Building Internet Firewalls, 2/e, ISBN: 1-56592-871-7 - E. D. Zwicky, et al - O'Reilly, 2000 8. CDMA Cellular Mobile Communications & Network Security, ISBN: 0-13-598418-1 - M. Y. Rhee, - Prentice Hall, 1998 9. Journal of Computer Security 10.ACM Transactions on Information and System Security 11. ACM Conference on Computer and Communications Security 12. IEEE Symposium on Security and Privacy 13. Internet documents - RFCs (Request for Comments) 14. Guide to Networking Essentials, Fourth Edition - Greg Tomsho, et al 15. Computer Networking Essentials - Debra Littlejohn Shinder 16. Networking Essentials: Hands-On, Self-Paced Training for Supporting Local and Wide Area Networks - Microsoft Corporation (Corporate Author) 17. Computer Network - A. S. Tanenbaum
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103040003]

Course: 304: Advanced Database Administration

<table>
<tr><td>Course Code</td><td>304</td></tr>
<tr><td>Course Title</td><td>Advanced Database Administration</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks/ Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2025</td></tr>
<tr><td>Purpose of Course</td><td>Understanding advanced database administration</td></tr>
<tr><td>Course Objective</td><td>To learn advanced database administration, database tuning and maintenance</td></tr>
<tr><td>Course Outcome</td><td>CO1 (Understand & Analyze): Explain the core concepts of Advanced Database Administration and analyze them from an industry-oriented perspective.
CO2 (Understand): Describe the internal architecture of the Oracle Database Management System, including its components and processes.
CO3 (Understand & Apply):
Explain the concepts of database security and demonstrate user management techniques in Oracle.
CO4 (Apply): Utilize various Oracle utilities for database administration tasks and operations.
CO5 (Apply & Evaluate): Implement practical techniques for database tuning and evaluate performance optimization strategies.
CO6 (Apply & Create): Perform backup and recovery operations and develop recovery strategies for different failure scenarios.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>RDBMS</td></tr>
<tr><td>Course Content</td><td>Unit 1: Oracle10g Instance creation and management, Database Architecture, and Concurrency Management
1.1 Oracle 10g Instance management
1.1.1 Oracle Instance
1.1.2 Installing Oracle
1.1.3 Oracle Optimal Flexible Architecture (OFA)
1.1.4 Locating initialization, listener.ora & sqlnet.ora files
1.1.5 Finding the alert log
1.1.6 Common environment variables
1.1.7 Structures in an Oracle Instance
1.1.8 Oracle Memory Structures, SGA and PGA
1.1.9 Oracle Processes and their purposes
1.1.10 Startup, nomount, mount and open database commands
1.2 Oracle10g Database Architecture
1.2.1 Oraclde10g management framework
1.2.2 Using the Database Creation Assistant (DBA)
1.2.3 Creating and dropping a database
1.2.4 Tablespaces
1.2.5 Tables and Indexes
1.2.6 Clusters
1.2.7 Partitioning of Tables and Indexes
1.2.8 Gathering and applying patches
1.3 Concurrency Management
1.3.1 Transactions, serialization, locks and latches</td></tr>
</table>

1.3.2 Lock modes
1.3.3 Detecting and resolving lock conflicts
1.3.4 Managing deadlocks

Unit 2: Interfacing with Oracle, Oracle*Net and Tablespace Management

2.1 Transaction management and PL/SQL
2.1.1 Oracle transaction management
2.1.2 Using SQL*Plus and iSQL*Plus
2.1.3 Using embedded Oracle with Pro*C & Java
2.1.4 PL/SQL & Triggers
2.1.5 Pining PL/SQL packages & compiling PL/SQL
2.1.6 System-level triggers – startup trigger, logon trigger, PL/SQL error trigger

2.2 Oracle*Net
2.2.1 Basic Network structure
2.2.2 Oracle*Net Files
2.2.3 Multi-threaded server
2.2.4 Create additional listeners
2.2.5 Create Oracle Net service aliases
2.2.6 Configure connect time failover
2.2.7 Oracle*Net names resolution

2.3 Tablespace Management
2.3.1 Tablespace management
2.3.1.1 Dictionary Managed Tablespaces
2.3.1.2 Locally Managed Tablespaces
2.3.1.3 Automatic Segment Space Management
2.3.1.4 Moving tablespaces online and offline
2.3.2 UNDO tablespace management
2.3.2.1 Use of undo segments
2.3.2.2 Creating an undo tablespace
2.3.2.3 User managed undo tablespaces
2.3.2.4 Automatic undo management
2.3.2.5 Monitor & Configure undo retention
2.3.2.6 Use the Undo Advisor
2.3.2.7 Size the undo tablespace

Unit 3: Oracle Utilities, Performance Tuning

3.1 Utilities
3.1.1 Datapump - Import/export
3.1.2 SQL*Loader
3.1.3 Oracle Streams
3.1.4 Automatic Database Diagnostic Monitor
3.1.5 Automatic Tuning Optimizer
3.1.6 Automatic Shared Memory Tuning

3.2 Oracle Performance Tuning
3.2.1 Locate invalid and unusable objects
3.2.2 Gather SQL optimizer statistics with dbms_stats
3.2.3 Basic Oracle performance metrics
3.2.4 Use OEM and dbms_alert to set warning and critical alert thresholds
3.2.5 The SQL Tuning Advisor
3.2.6 The SQL Access Advisor
3.2.7 Interpreting server generated alerts
3.2.8 Oracle advisory utilities vdb_cache_advice, vshared pool_advice, v$pga_aggregate_target_advice
3.2.9 Using OEM performance screens

	3.2.10 Fixing performance issues **Unit 4: User Management, Security, Backup & Recovery** 4.1 User management 4.1.1 Creating Users 4.1.2 Altering users 4.1.3 User Profiles 4.1.4 User resource groups 4.1.5 Granting privileges & roles 4.1.6 Auditing user activity with dbms_audit 4.2 Oracle Security 4.2.1 Password use in Oracle, Password encryption and password aging, External authentication, Using Single sign-on (SSO) 4.2.2 Object security 4.2.3 Virtual Private Databases (VPD) in Oracle 4.2.4 Oracle "grant execute" security 4.2.5 Use of Roles in Oracle 4.2.6 Register for security updates 4.3 Backup & Recovery 4.3.1 Oracle backup & recovery planning 4.3.2 Parallel instance recovery 4.3.3 Basics of checkpoints, redo log files, and archived log files 4.3.4 Using ARCHIVELOG mode 4.3.5 Creating consistent Oracle backups 4.3.6 Online hot backups 4.3.7 Incremental Oracle backups 4.3.8 Automating database backups with dbms_scheduler 4.3.9 Monitor the flash recovery area 4.3.10 Recovering from loss of a Control file 4.3.11 Recovering from loss of a Redo log file 4.3.12 Recovering from loss of a system-critical data file 4.3.13 Recovering from loss of a non system-critical data file
Reference Books	1. Essentials: Oracle Database 10g by Rick Greenwald, Robert Stackowiak, Jonathan Stern, O'Reilly 2. Oracle High Performance Tuning for 9i and 10g by Gavin Powell, Digital Press 3. Oracle Database 10g, DBA Handbook by Loney, Kevin, Bryla, Bob, Oracle Press 4. Oracle Database 10g - The Complete Reference by Loney, Kevin, Oracle Press 5. Oracle Database 10g: A Beginner's Guide by Micheal Abbey, Ian Abramson Osborne, Oracle Press Series
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment is based on class attendance, participation, class test, quiz, assignment, seminar, internal examination etc. 70% assessment is based on semester end University External examination

[Subject code-2111040103050002]

Course: 305: Open Source Web Based Programming

Course Code	305
Course Title	**Open Source Web Based Programming**
Credit	4
Teaching per Week	4 Hrs.
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to understand fundamentals of Open-Source web-based Programming. The course also imparts students learning about Open-source web-based scripting language PHP and MySQL database. It also includes MVC or Three tier architecture of web-based programming.
Course Objective	1. Identify and recall the fundamental concepts, tools, and terminologies used in open-source web technologies. 2. Explain the architecture and working principles of various open-source web technologies and frameworks. 3. Develop basic web applications using open-source tools and integrate essential components like forms, databases, and APIs. 4. Differentiate between various open-source frameworks (e.g., ReactJS, Node.js, etc.) and analyze their advantages and limitations in specific use cases. 5. Design and build a fully functional, database-driven web application using modern open-source technologies, adhering to best practices in structure, security, and UI/UX.
Course Outcome	CO1 (Remembering / Understanding): Recognize the fundamentals of PHP and MySQL, their installation and configuration, and identify language features such as operators, variables, control structures, and error handling. CO2 (Applying): Apply PHP functions to perform string manipulation, array operations, mathematical calculations, and handle date/time operations, while also using object-oriented features like classes, objects, constructors, and inheritance. CO3 (Applying): Establish and manage MySQL databases, execute CRUD operations, handle server-side communication using PHP and AJAX, and implement form validation and real-time user feedback in dynamic web applications. CO4 (Analyzing): Analyze and integrate XML for data representation and web services for remote communication, utilizing AJAX for asynchronous requests and dynamic content handling in web applications. CO5 (Applying / Creating): Develop full-stack web applications using the Laravel framework by applying MVC architecture, managing routing, controllers, and views, handling form submissions, and ensuring application security with features like CSRF, XSS, and SQL injection prevention. CO6 (Applying / Creating): Implement advanced concepts in Laravel such as dynamic UI rendering with Blade, email communication, user authentication, role-based access control, and data encryption, while ensuring secure data handling and applying best practices in application security.
Mapping between COs with PSOs	

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■							
CO2	■		■					
CO3	■	■	■		■			
CO4		■	■	■		■	■	■

	CO5 ■ ☐ ■ ☐ ■ ■ ☐ ☐ CO6 ■ ■ ☐ ■ ☐ ■ ■ ■
Pre-requisite	Knowledge of HTML and SQL
Course Content	**Unit 1 : Introduction to Open source Web based Programming** 1.1 Introduction to PHP & MySql 1.2 Installation of PHP and MySql 1.3 Language Characteristics & Features 1.4 Operators and Variables, Control Structures, Looping and Error handling 1.5 PHP functions 1.5.1 String Functions 1.5.2 Array Functions 1.5.3 Mathematical Functions 1.5.4 Graphics Library (GD Support) 1.5.6 Date and Time Functions 1.5.7 Misc. Function 1.6 State management Techniques 1.7 Object Oriented Features of PHP 1.7.1 Classes and Objects 1.7.2 Use of constructors 1.7.3 Serialization 1.7.4 Inheritance **Unit 2: Advanced Server-Side Integration and Dynamic Web Communication** 2.1 MySQL Server Setup and Management 2.1.1. Installation and Configuration of MySQL Server 2.1.2. Creating and Managing Tables 2.1.3. Displaying Databases and Managing User Access 2.1.4. Establishing Database Connections and Data Handling using Server-Side Scripts 2.2 AJAX-Based Web Communication 2.2.1 Fundamentals of HTTP Requests and Responses 2.2.2 The XMLHttpRequest Object: Methods and Properties 2.2.3 Handling Cross-Browser AJAX Implementations 2.2.4 Sending Asynchronous Requests to the Server 2.2.5 Integrating PHP with AJAX for Client-Server Communication 2.2.6 Server-Side Processing and Dynamic Content Handling 2.2.7 Form Validation and Real-Time User Feedback 2.2.8 Executing AJAX-Driven Database Queries 2.3 Structured Data Exchange and Services 2.3.1 Introduction to XML for Data Representation 2.3.2 Basics of Web Services for Remote Communication and Integration **Unit 3: Full-Stack Web Development with Laravel Framework** 3.1 MVC Architecture & Laravel Introduction 3.1.1 Fundamentals of MVC (Model-View-Controller) Architecture 3.1.2 Overview of Laravel and its Core Concepts 3.2 Application Structure & Routing 3.2.1 Laravel Application Workflow 3.2.2 Route Definitions and URL Mapping 3.2.3 Middleware for Request Filtering 3.3 Request Lifecycle & Controllers 3.3.1 Handling HTTP Requests and Responses 3.3.2 Creating and Managing Controllers 3.4 Views & Dynamic Content Rendering 3.4.1 Blade Templating Engine for View Management

	3.4.2 URL Generation and Layout Inheritance 3.5 Form Handling & User Input 3.5.1 Form Submission and File Upload Mechanisms 3.5.2 Input Validation Techniques 3.5.3 Session Management and Cookie Usage 3.6 Error Management & Application Security 3.6.1 Exception Handling and Custom Error Pages 3.6.2 Security Features in Laravel (CSRF, XSS, SQL Injection Prevention) 3.7 Database Integration 3.7.1 Connecting to Databases Using Eloquent ORM 3.7.2 Performing CRUD Operations and Data Relationships **Unit 4 : Advanced Concepts in MVC with Laravel** 4.1 Dynamic UI Rendering with Blade 4.1.1 Advanced Blade Templating Techniques 4.1.2 Layout Inheritance, Components, and Slots 4.1.3 Conditional and Loop Directives 4.2 Email Communication in Laravel 4.2.1 Sending Emails using Laravel Mail 4.2.2 Email Templates and Queueing Mails 4.2.3 SMTP Configuration and Mail Drivers 4.3 User Authentication 4.3.1 Built-in Authentication Scaffolding 4.3.2 Login, Registration, and Password Reset 4.3.3 Customizing Authentication Logic 4.4 Access Control and Authorization 4.4.1 Role-Based Access Control (RBAC) 4.4.2 Gates and Policies for Fine-Grained Access 4.4.3 Middleware for Route Protection 4.5 Data Security and Encryption 4.5.1 Encrypting and Decrypting Data 4.5.2 Storing Secure Information 4.5.3 Laravel's Cryptographic Tools and Best Practices
Reference Books	1. Beginning PHP, Apache, MySQL Web Development - Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass, Gary Mailer Wrox Publication 2. Professional PHP Programming - Jesus Castagnetto ,Wrox Press Ltd 3. Beginning PHP and MySQL: From Novice to Professional - W. Jason Gilmore, Apress 4. Php: The Complete Reference - Steven Holzner, Tata Mcgraw Hill Education Private Limited 5. AJAX and PHP: Building Responsive Web Applications - Bogdan Brinzarea, CristianDarie packtpub 6. Php manual www. Php.com 7. Beginning Laravel Build Websites with Laravel 5.8 - Sanjib Sinha · 2019 - Apress 8. Laravel The Ultimate Beginner's Guide to Learn Laravel Step by Step, 2nd Edition - Mem Lnc, Rufus Stewart
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment

[Subject code-2111040103060002]

Course: 306: Programming Skills VIII

Course Code	306
Course Title	**Programming Skills VIII**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	Learn practically Embedded systems and IOT / ML
Course Objective	Learn Embedded systems and IOT practically, understand the working of Micro-Controller & Micro-Computer and using various types of sensors and its Programming. / Implement various algorithms used in Machine learning and introduce students with Programming in ML.
Pre-requisite	C/C++ Language / Python Programming
Course Outcome	After successful completion, students will be able to work with different types of Micro-Controllers, Micro-Computers and sensors for their IOT based application development. / After successful completion, students will be able to work with different types of ML algorithms and ML based application development.
Course Content	Practical based on paper no 301. (Embedded systems and IOT/ML) Separate journal to be prepared for this Course
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103070002]

Course: 307: Programming Skills IX

Course Code	307
Course Title	**Programming Skills IX**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to understand the Networking and its security practically
Course Objective	Learn how to implement Network and its security practically.
Pre-requisite	Network fundamentals
Course Outcome	After studying the course, students will be able to practically implement network and its security
Course Content	Practical based on paper no 303. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Exercises, Self-study, and/or Assignment

Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103080002]

Course: 308: Programming Skills X

Course Code	308
Course Title	**Programming Skills X**
Credit	2
Teaching per Week	2 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to learn network administration
Course Objective	Students will learn administration of network practically
Pre-requisite	Networking fundamentals
Course Outcome	After studying the course, students will be able to practically perform administrative tasks of networks practically
Course Content	Practical based on paper no 304. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Exercises (to be documented in a separate journal), Self-study, and/or Assignment
Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2111040103090002]

Course: 309: **Programming Skills XI**

Course Code	309
Course Title	**Programming Skills XI**
Credit	3
Teaching per Week	3 Hrs.
Minimum weeks/ Semester	15 (Including Lab. work, examination, preparation, holidays etc.)
Review / Revision	June 2025
Purpose of Course	This course helps students to implement the basic and advanced concepts of PHP/MySql practically.
Course Objective	Learning to develop and deploy websites using PHP/MySql practically.
Pre-requisite	Basic scripting, programming, html.
Course Outcome	After studying the course, students will be able to practically develop dynamic websites using PHP/MySql.
Course Content	Practical based on paper no 305. Separate journal to be prepared for this Course.
Reference Books	-----
Teaching Methodology	Lab work, Practical Programming Exercises (to be documented in a separate journal), Self-study, and/or Assignment

Evaluation Method	30% Internal assessment is based on Lab attendance, practical test, practical internal examination etc. 70% assessment is based on semester end University External practical examination

[Subject code-2211040104010002]

MCA 4th Semester

Course: 401: **Seminar**

Course Code	401
Course Title	**Seminar**
Credit	6
Review / Revision	June 2025

- The students are required to prepare a seminar on a relevant topic concerning the subject of interest of the student; as well as latest technology.
- The students must prepare documentation of the seminar.
- At the end of the semester, the students have to submit the seminar reports in spiral bounded form to the institution.
- Seminar Completion Certificate issued by the institute is mandatory for appearing in Seminar Presentations.
- The Seminar Presentation will be conducted as per the University exam schedule.
  The students have to submit the following reports at the institution:

  1. Seminar Topic Chosen
  2. Institution Certificate for Seminar

[Subject code-2211040104020002]

Course: 402: **Project**

Course Code	402
Course Title	**Project**
Credit	24
Review / Revision	June 2025

- The students are required to carry out full time software development project in a company.
- The students must prepare documentation of the project completed as per the guidelines given by the institute.
- At the end of the semester, the students have to submit the project reports in bounded form to the institution.
- Project Completion Certificate issued by the institute is mandatory for appearing in Project Presentation and Viva – Voce.
- The Project Presentation and Viva – Voce will be conducted as per the University exam schedule.

  The students have to submit the following reports at the institution:

  1. Project Joining Report
  2. Progress Reports
  3. Project Completion Certificate from the company
  4. Institution Certificate for Project
  5. Non-disclosure of Source Code Certificate (In case the student is unable to submit project source code)