

ANDHRA KESARI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MASTER OF COMPUTER APPLICATION (MCA) COURSE

2026-27

AK UNIVERSITY – MCA Course Structure

MCA I Year – I Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MCA26101	C	Data Structures with C++	30	70	3	1	0	4
2	MCA26102	C	Database Management Systems	30	70	3	1	0	4
3	MCA26103	C	Operating Systems	30	70	3	1	0	4
4	MCA26104	CF	Probability and Statistics	30	70	3	1	0	4
5	MCA26105A	EF-I	1. Computer Organization	30	70	3	1	0	4
	MCA26105B		2. Digital Logic Design						
	MCA26105C		3. Discrete Mathematical Structures						
6	MCA26106	P	Data Structures lab	25	50	0	0	6	3
7	MCA26107	P	Database Management Systems Lab	25	50	0	0	6	3
8	MCA26108	SEC	Communication Skills	50	-	0	4	0	2
			TOTAL	**260**	**490**	**15**	**9**	**12**	**28**

MCA I Year – II Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MCA26201	C	Software Engineering	30	70	3	1	0	4
2	MCA26202	C	Computer Networks	30	70	3	1	0	4
3	MCA26203	C	Web Technologies	30	70	3	1	0	4
4	MCA26204	CF	Research Methodology & Intellectual Property Rights (IPR)	30	70	3	1	0	4
5	MCA26205A	EF-II	1. Programming and Problem-Solving using Python	30	70	3	1	0	4
	MCA26205B		2. Introduction to R Programming						
6	MCA26206	P	Python Programming / R Programming Lab	25	50	0	0	6	3
7	MCA26207	P	Web Technologies Lab	25	50	0	0	6	3
8	MCA26208	SEC	Seminar	50	-	0	4	0	2
9	MCA26209	MC	SWAYAM Courses	-	-	0	0	0	2*
			TOTAL	**260**	**490**	**15**	**9**	**12**	**28**

C – Mandatory Core
CF – Compulsory Foundation
P – Practical
EF – Elective Foundation
SEC – Skill Enhancement Course
MC – Mandatory Courses (MOOCs)

*__NOTE:__ The credits assigned for SWAYAM Course (MOOCs Mandatory Course) will not be counted for the GPA.

MCA II Year – III Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MCA26301	C	Data Mining and Big Data	30	70	3	1	0	4
2	MCA26302	C	Cloud Computing	30	70	3	1	0	4
3	MCA26303	C	Artificial Intelligence & Machine Learning	30	70	3	1	0	4
4	MCA26304A	E-I	1. Blockchain Technology	30	70	3	1	0	4
	MCA26304B		2. Cryptography & Network Security						
	MCA26304C		3. Cyber Security						
5	MCA26305A	E-II	1. WISER – Quantum Foundation and Applied Technologies	30	70	3	1	0	4
	MCA26305B		2. Mobile Computing with Android						
	MCA26305C		3. Microsoft Dynamics						
6	MCA26306	P	Data Mining and Big Data Lab	25	50	0	0	6	3
7	MCA26307	P	Cryptography & Network Security lab	25	50	0	0	6	3
8	MCA26308	SEC	Technical Report Writing	50	-	0	4	0	2
9	MCA26309	MC	SWAYAM Plus Courses	-	-	0	0	0	2*
			TOTAL	**260**	**490**	**15**	**9**	**12**	**28**

MCA II Year – IV Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MCA26401	SEC	Project Work	100	200	0	0	36	12
			TOTAL	**100**	**200**	**0**	**0**	**36**	**12**

C – Mandatory Core
CF – Compulsory Foundation
P – Practical
EF – Elective Foundation
SEC – Skill Enhancement Course
MC – Mandatory Courses (MOOCs)

*__NOTE:__ The credits assigned for SWAYAM Plus Courses (MOOCs Mandatory Course) will not be counted for the GPA.

SEMESTER III

Semester	MCA26301: DATA MINING AND BIG DATA	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	To identify the scope and essentiality of Data Warehousing and Mining.
LO2	To analyze data, choose relevant models and algorithms for respective applications
LO3	To develop research interest towards advances in data mining
LO4	To provide an overview of an exciting growing field of big data analytics.

COURSE OUTCOMES (CO):

Students upon completion of this course will be able to:

CO1	Understand Data Warehouse fundamentals, Data Mining Principles
CO2	Identify appropriate data mining algorithms to solve real world problems
CO3	Compare and evaluate different data mining techniques like classification, prediction, clustering and association rule mining.
CO4	Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.
CO5	Acquire fundamental enabling techniques and scalable algorithms like Hadoop, Map Reduce in big data analytics.

Course Content:

UNIT - I

Data Warehouse and OLAP Technology: An Overview: What is Data Warehouse? - A Multidimensional Data Model - Data warehouse Architecture - From Data Warehousing to Data Mining

Data mining - Introduction, Data mining on what kind of data , Data mining functionalities, classification of Data mining systems, Major issues in Data mining

UNIT - II

Mining Association rules in large databases - Association rule mining, Mining single-Dimensional Boolean association rules from Transactional databases, Mining multi-Dimensional Association rules from relational Databases and Data Warehouses

Classification and Prediction - Introduction classification by decision tree induction, Bayesian Classification. Other classification methods, classification by back propagation, Prediction, classifier accuracy

UNIT - III

Cluster analysis - Introduction, Types of data in cluster analysis, a categorization of major clustering methods, portioning methods, hierarchical methods, Density based methods: DBSCAN, Grid-based method : STING , Model based clustering method: Statistical Approach, outlier analysis.

UNIT - IV

Big Data: Introduction - distributed file system - Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Big data applications.

Hadoop: Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands, Anatomy of File Write and Read., Hadoop MapReduce paradigm. Writing Hadoop MapReduce Programs

PRESCRIBED TEXTBOOKS:

1. Jiawei Han Micheline Kamber, “Data mining & Techniques”, Morgan Kaufmann publishers
2. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, “Professional Hadoop Solutions”, Wiley, ISBN: 9788126551071, 2015.
3. Chris Eaton, Dirk deroos et al., “Understanding Big data ”, McGraw Hill, 2012.
4. Tom White, “HADOOP: The definitive Guide” , O Reilly 2012.

Semester	MCA26302: CLOUD COMPUTING	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	The student will learn about the cloud environment, building software systems and components that scale to millions of users in modern internet
LO2	cloud concepts capabilities across the various cloud service models including Iaas, Paas, Saas,
LO3	developing cloud-based software applications on top of cloud platforms.

COURSE OUTCOMES (CO):

Students successfully completing this module will be able to:

CO1	Understanding the key dimensions of the challenge of Cloud Computing
CO2	Assessment of the economics, financial, and technological implications for selecting cloud computing for own organization

Course Content:

UNIT - I

Cloud Computing Basics: Cloud Computing Overview, applications, Intranets and the Cloud, First Movers in the Cloud.

Your Organization and Cloud Computing: When you can use Cloud Computing, Benefits Limitations.

Cloud Computing with Titans: Google, EMC, NetApp, Microsoft, Amazon, Salesforce.com, IBM and partnerships.

UNIT -II

The Business Case for Going to the Cloud - Cloud Computing Services, How those Applications Help your Business, Deleting your Datacenter, Thomson Reuters.

Hardware and Infrastructure: Clients, Security- Data Leakage, Offloading work, Logging, Forensics, Compliance VPNs, Key management; Network; Services - identify, integration, mapping, payment, search.

Accessing the Cloud: Platforms web applications, Web APIs, Web Browsers.

UNIT- III

Cloud Storage: Overview, Cloud storage providers.

Software as a Service: Overview, Driving Forces, Company Offerings and Industries.

Software plus Services: Overview, Mobile Device Integration, Providers, Microsoft online.

UNIT-IV

Local Clouds, Thin Clients, Thick clients: Types of Virtualizations, Virtualization in Your Organization, Server Solutions, Thin Clients.

Migrating to the Cloud: Cloud Services for Individuals, Cloud Services Aimed at the Mid-Market, Enterprise-Class Cloud Offerings, Migration

Best Practices and the Future of Cloud Computing - Analyze Your Service, Best Practices, How Cloud Computing Might Evolve.

PRESCRIBED TEXTBOOK:

Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, “Cloud Computing A Practical Approach”, Mc Graw-Hill.

REFERENCE BOOKS:

1. Michael Miller, “Cloud Computing”, Pearson Education, New
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, "Mastering Cloud Computing", Mc Graw Hill Education.
3. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pvt. Limited, July 2008.
4. Cloud Application Architectures, George Reese, ISBN: 8184047142, Shroff/O' Reilly, 2009.

Semester	MCA26303: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	L	T	P	C
III		3	1	0	4

COURSE DESCRIPTION

This course introduces the principles and techniques of Artificial Intelligence and Machine Learning. It covers intelligent problem-solving, search, knowledge representation, reasoning, planning, and learning from data. Students study representative learning algorithms, methods for evaluating predictive models, and applications of intelligent systems.

LEARNING OBJECTIVES (LO):

1. LO1: Introduce the foundations of Artificial Intelligence and intelligent problem-solving.
2. LO2: Develop an understanding of knowledge representation, reasoning, and planning.
3. LO3: Explain the principles of concept learning and representative machine learning algorithms.
4. LO4: Introduce methods for evaluating hypotheses and understanding learning behaviour.
5. LO5: Familiarize students with evolutionary computation, reinforcement learning, and applications of intelligent systems.

COURSE OUTCOMES (CO):

Upon successful completion of the course, students will be able to:

- **CO1:** Formulate problems such as state-space searches and apply appropriate search techniques.
- **CO2:** Represent knowledge using logic, rules, semantic networks, and frames, and explain reasoning and planning methods.
- **CO3:** Apply concept learning, decision tree, Bayesian, and instance-based learning techniques to suitable problems.
- **CO4:** Evaluate learning models and explain fundamental concepts of computational learning theory.
- **CO5:** Explain genetic algorithms, reinforcement learning, and the integration of prior knowledge into learning systems.

Course Content:

UNIT I: FOUNDATIONS OF ARTIFICIAL INTELLIGENCE AND PROBLEM-SOLVING

Introduction to Artificial Intelligence: Meaning and scope of AI; AI problems; underlying assumptions; AI techniques; levels of modelling; criteria for success; relationship between Artificial Intelligence and Machine Learning.

Problems, Problem Spaces, and Search: Defining a problem as a state-space search; production systems; problem characteristics; production-system characteristics; issues in the design of search programs; breadth-first and depth-first search.

Heuristic Search Techniques: Generate-and-test; hill climbing; best-first search; problem reduction; constraint satisfaction; means–ends analysis.

Planning: Components of a planning system; the Blocks World; goal-stack planning; nonlinear planning using constraint posting; introduction to hierarchical planning and reactive systems.

UNIT II: KNOWLEDGE REPRESENTATION, REASONING, AND AI APPLICATIONS

Knowledge Representation: Representations and mappings; approaches to knowledge representation; issues in knowledge representation; the frame problem; semantic networks and frames.

Predicate Logic and Rule-Based Representation: Representing facts, instance relationships, and "is-a" relationships; computable functions and predicates; resolution; natural deduction; procedural and declarative knowledge; logic programming; forward and backward reasoning; matching and control knowledge.

Reasoning under Uncertainty: Introduction to nonmonotonic reasoning; nonmonotonic logics; implementation issues; augmenting a problem solver; distinction between nonmonotonic and probabilistic reasoning.

Natural Language Processing and Commonsense Reasoning: Syntactic processing; semantic analysis; discourse and pragmatic processing; introduction to qualitative physics, commonsense ontologies, and memory organization.

Expert Systems: Representation and use of domain knowledge; expert-system architecture; expert-system shells; explanation facilities; knowledge acquisition.

UNIT III: FOUNDATIONS OF MACHINE LEARNING AND SUPERVISED LEARNING

Introduction to Machine Learning: Well-posed learning problems; designing a learning system; perspectives and issues in machine learning.

Concept Learning: A concept-learning task; concept learning as search; general-to-specific ordering; Find-S algorithm; version spaces; Candidate-Elimination algorithm; limitations of version-space methods; inductive bias.

Decision Tree Learning: Decision tree representation; appropriate learning problems; basic decision tree learning algorithm; hypothesis-space search; inductive bias; issues in decision tree learning, including overfitting and handling different types of attributes.

Bayesian Learning: Bayes' theorem and concept learning; maximum-likelihood hypotheses; least-squared-error hypotheses; predicting probabilities; minimum description length principle; Bayes-optimal classifier; Gibbs algorithm; Naïve Bayes classifier; text classification; introduction to Bayesian belief networks and the Expectation–Maximization algorithm.

Evaluating Learning Models: Estimating hypothesis accuracy; basics of sampling theory; confidence intervals; comparing the errors of two hypotheses; comparing learning algorithms; training and test data; introductory model-validation practices.

UNIT IV: ADVANCED LEARNING METHODS AND INTELLIGENT ADAPTATION

Computational Learning Theory: Probably Approximately Correct learning; sample complexity for finite hypothesis spaces; introduction to sample complexity for infinite hypothesis spaces; mistake-bound model of learning.

Instance-Based Learning: k-nearest neighbor learning; locally weighted regression; radial basis functions; case-based reasoning; lazy and eager learning.

Genetic Algorithms: Motivation; representation of candidate solutions; fitness evaluation; selection, crossover, and mutation; illustrative examples; hypothesis-space search; introduction to genetic programming; models of evolution and learning; parallel genetic algorithms.

Combining Inductive and Analytical Learning: Motivation; inductive–analytical approaches; using prior knowledge to initialize hypotheses, alter search objectives, and augment search operators.

Reinforcement Learning: The learning task; rewards and actions; Q-learning; learning in nondeterministic environments; temporal-difference learning; generalization from examples; relationship to dynamic programming.

PRESCRIBED TEXTBOOKS

1. Elaine Rich and Kevin Knight, *Artificial Intelligence*, Second Edition, Tata McGraw-Hill, 1991.
2. Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

REFERENCE BOOKS

1. Michael Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, Second Edition, Pearson Education.
2. Patrick Henry Winston, *Artificial Intelligence*, Third Edition, Addison-Wesley, 1993.
3. Ethem Alpaydin, *Introduction to Machine Learning*, PHI Learning.
4. Stephen Marsland, *Machine Learning: An Algorithmic Perspective*, Taylor & Francis.

Semester	MCA26304A: BLOCKCHAIN TECHNOLOGY	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	To introduce the concept of Blockchain
LO2	To overcome the problems of centralization
LO3	To introduce the concept of Bitcoin
LO4	To make them familiar with Bitcoin network, payments, clients and APIs

COURSE OUTCOMES (CO):

Students successfully completing this module will be able to:

CO1	Explain the fundamental concepts of blockchain technology, including decentralization, distributed systems, types of blockchains, and the evolution of block chain and Bitcoin.
CO2	Analyze cryptographic principles such as symmetric and asymmetric cryptography, public–private key mechanisms, and their role in securing block chain transactions and financial systems.
CO3	Describe the architecture and working of Bitcoin, including digital wallets, transactions, mining process, alternative cryptocurrencies, and the limitations of Bitcoin.
CO4	Apply knowledge of Bitcoin networks, payment mechanisms, wallets, clients, and APIs to understand real-world blockchain-based financial applications.

Course Content:

UNIT – I

Blockchain, the growth of blockchain technology, distributed systems, the history of blockchain and Bitcoin, types of blockchains, Decentralization, methods of decentralization, routes of decentralization, blockchain and full ecosystem decentralization, smart contracts, Decentralized organizations and platforms for decentralization.

UNIT – II

Symmetric Cryptography, working with the OpenSSL command line, cryptographic primitives. Public Key Cryptography, asymmetric cryptography, public and private keys and financial markets and trading.

UNIT – III

Introducing Bitcoin, Bitcoin, digital keys and addresses, transactions, blockchain, mining. Alternative Coins. Limitations of Bitcoin

UNIT – IV

Bitcoin Network and payments, The Bitcoin network, wallets, Bitcoin payments, innovation in Bitcoin, Bitcoin Clients and APIs.

PRESCRIBE TEXTBOOK

Mastering Blockchain 2nd Edition, Imran Bashir, PACKT Publication

REFERENCE BOOKS

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.

Semester	MCA26304B: CRYPTOGRAPHY AND NETWORK SECURITY	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	Security breaches can be very expensive in terms of business disruption and the financial losses that may result.
LO2	Increasing volumes of sensitive information are transferred across the internet or intranets connected to it.
LO3	Networking that makes use of internet links are becoming more popular because they are cheaper than dedicated leased lines. This, however, involves different users sharing internet links to transport their data.
LO4	Directors of business organizations are increasingly required to provide effective information security.

COURSE OUTCOMES (CO):

Students successfully completing this module will be able to:

CO1	Identify some of the factors driving the need for network security.
CO2	Identify and classify examples of attacks.
CO3	Compare and contrast symmetric and asymmetric encryption systems and their vulnerability to attack, and explain the characteristics of hybrid systems
CO4	Describe the use of hash functions and explain the characteristics of one-way and collision- free functions.
CO5	Describe and distinguish between different mechanisms to assure the freshness of a message
CO6	Explain the role of third-party agents in the provision of authentication services.
CO7	Discuss the effectiveness of passwords in access control and the influence of human behavior
CO8	Identify types of firewall implementation suitable for differing security requirements
CO9	Distinguish between firewalls based on packet-filtering routers, application-level gateways and circuit level gateways.

Course Content:

UNIT-I

Introduction: Security trends, the OSI security architecture, security attacks, security services, security mechanisms, a model for network security.

Classical encryption techniques: Symmetric cipher model, Substitution techniques, Transposition techniques, Rotor machines, Steganography.

Block cipher and the data encryption standard: Block cipher principles, the strength of DES, Differential and linear cryptanalysis, Block cipher design principles.

Confidentiality using Symmetric Encryption: Placement of encryption function, Traffic confidentiality, key distribution, random number generator.

UNIT-II

Public key cryptography and RSA: Principles of public key crypto systems, The RSA algorithm

Key management: Other public-key crypto systems: Key management, Diffie-Hellman key exchange.

Message authentication and hash functions: Authentication requirements, Authentication functions, message authentication codes, Hash functions, security of hash functions and MACs.

Digital signatures and authentication protocols: Digital signatures, Authentication protocols, Digital Signature standard.

UNIT-III

Authentication Applications: Kerberos, X.509 authentication service

Email Security: Pretty good privacy, S/MIME

IP security: IP security overview, IP security architecture, Authentication header, Encapsulating security payload, combining security associations, key management.

Web security: Web security considerations, Secure Socket Layer and transport layer security, Secure electronic transaction.

UNIT-IV

Intruders: Intruders, Intrusion detection, password management

Malicious Software: Viruses and related threads, virus counter measures, distributed denial of service attacks.

Firewalls: Firewall Design principles, trusted systems, common criteria for information technology, security evaluation.

PRESCRIBED TEXTBOOK:

William Stallings, “Cryptography and Network Security”, Fourth edition, PHI.

REFERENCE BOOKS:

1. William Stallings, “Network Security Essentials - Applications and Standards”, Third Edition, Pearson Education (2007).
2. Chris McNab, “Network Security Assessment”, 2nd Edition, OReilly (2007).
3. Jon Erickson, “Hacking - The Art of Exploitation”, SPD, NOSTARCH Press (2006).
4. Neal Krawety, “Introduction to Network Security”, Thomson (2007).

Semester	MCA26304C: CYBER SECURITY	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The students will be able to:

LO1	Understand various block cipher and stream cipher models
LO2	Describe the principles of public key crypto systems, hash functions and digital signature
LO3	To get a firm knowledge on Cyber Security Essentials

COURSE OUTCOMES (CO):

At the end of the course, the students will be able to:

CO1	Explain basic security algorithms required by any computing system
CO2	Analyze the possible security attacks in complex real time systems and their effective Counter measures
CO3	Compare various governing bodies of cyber laws
CO4	Infer various privacy policies for an organization
C05	Solve the vulnerabilities in any computing system and hence be able to design a security solution

Course Content:

UNIT – I

Introduction to Security: Data Encryption Standard, Block cipher principles, block cipher modes of operation, Advanced Encryption Standard (AES), Triple DES, Blowfish, RC5 algorithm

Public Key Cryptography and Hash Algorithms: Principles of public key crypto systems, The RSA algorithm, Key management, Diffie Hellman Key exchange, Hash functions, Hash Algorithms (MD5, Secure Hash Algorithm)

UNIT – II

Fundamentals of Cyber Security: Attacker Techniques and Motivations: How Hackers Cover Their Tracks, Fraud Techniques, Threat Infrastructure.

Exploration: Techniques to Gain a Foot hold (Shell code, SQL Injection, Malicious PDF Files), Misdirection, Reconnaissance, and Disruption Methods

UNIT – III

Privacy to provide Cyber Security: Privacy Concepts, Privacy Principles and Policies, Authentication and Privacy, Data Mining, Privacy on the Web, Email Security, Privacy Impacts of Emerging Technologies.

UNIT – IV

Cyber Security Management and Incidents: Security Planning, Business Continuity Planning, Handling Incidents, Risk Analysis, Dealing with Disaster.

Legal Issues and Ethics: Protecting programs and Data, Information and the law, Rights of Employees and Employers.

Emerging Topics: The Internet of Things, Cyber Warfare.

PRESCRIBE AND REFERENCE TEXTBOOKS

1. William Stallings, “Cryptography and Network Security”, Pearson Education, 6th Edition, 2013.

2. Graham Howard, R., Olson Cyber Security Essentials, CRC Press, 2011.

3. Charles P. P. Fleeger Shari Lawrence P. Fleeger Jonathan Margulies, Security in Computing, 5th Edition, Pearson Education, 2015.

Semester	MCA26305A: QUANTUM FOUNDATION AND APPLIED TECHNOLOGIES	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	Introduce the fundamental principles and concepts of quantum science and build a strong foundation in quantum technologies.
LO2	Develop the ability to analyze and differentiate classical and quantum systems based on the fundamental principles of quantum mechanics.
LO3	Provide exposure to quantum algorithms, circuits, protocols, software platforms, real quantum hardware and quantum research laboratory environments.
LO4	Familiarize students with quantum simulators, noise models, variational quantum algorithms, high-performance computing, artificial intelligence and scalable quantum computing systems.
LO5	Prepare students for academic, research and industry opportunities in quantum technologies and entrepreneurship.

COURSE OUTCOMES (CO):

Students successfully completing this module will be able to:

CO1	Explain the fundamental concepts, evolution and significance of quantum science and quantum technologies.
CO2	Differentiate classical and quantum systems using concepts such as qubits, superposition, entanglement and quantum measurement.
CO3	Describe the principles, components and applications of quantum computing, sensing, communication, networking and hardware systems.
CO4	Construct and interpret basic quantum circuits and explain key algorithms, simulators, noise models and real hardware workflows.
CO5	Analyze quantum applications in industry and demonstrate awareness of HPC, AI integration, scalable systems, careers, innovation and entrepreneurship.

Course Content:

UNIT – I: FOUNDATIONS OF QUANTUM SCIENCE AND TECHNOLOGIES

Introduction to quantum science, its evolution and significance in modern technology; quantum systems, qubits and quantum information; basic mathematical terminology used in quantum computing; comparison of classical and quantum computing paradigms; overview of the quantum ecosystem, emerging technologies, future research directions, quantum sensing, quantum annealing and selected applications in chemistry and materials science.

UNIT – II: QUANTUM COMMUNICATION, NETWORKING AND HARDWARE SYSTEMS

Quantum communication, quantum networking and the quantum internet; entanglement-based communication, quantum data transmission and network architectures; applications in power and energy systems; fundamentals of quantum hardware, superconducting systems, cryogenic infrastructure and dilution refrigeration; exposure to quantum research laboratories and hardware environments.

UNIT – III: QUANTUM COMPUTING FUNDAMENTALS AND ALGORITHMS

Global landscape of quantum computing; DiVincenzo criteria, qubits and core quantum principles; quantum states, measurement, superposition, entanglement, two-qubit gates, GHZ states, superdense coding and quantum teleportation; quantum protocols, circuits and algorithms; IBM Qiskit for creating and visualising circuits; Quantum Fourier Transform, Quantum Phase Estimation, Grover's Algorithm and introductory QAOA applications.

UNIT – IV: QUANTUM HARDWARE, NOISE AND VARIATIONAL COMPUTING

Quantum simulators, noise modelling and execution of algorithms on real platforms; trapped-ion quantum computers and high-fidelity qubit control; variational quantum algorithms, parameterized quantum circuits, optimization techniques and the variational computing framework; ZX Calculus and emerging applications in scientific computing and computational fluid dynamics.

UNIT – V: QUANTUM ECOSYSTEM, HPC, AI AND ENTREPRENEURSHIP

Scalable quantum computing, hybrid architectures and resource estimation; HPC, AI and integration with quantum technologies; fault-tolerant quantum systems, benchmarking and evaluation of hardware and algorithms; applications in weather modelling, optimization, energy and chemistry; global quantum ecosystem, workforce needs, careers, innovation and entrepreneurship.

PRESCRIBE AND REFERENCE TEXTBOOKS

1. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information.
2. Robert S. Sutor, Dancing with Qubits: How Quantum Computing Works and How It Can Change the World.
3. Phillip Kaye, Raymond Laflamme and Michele Mosca, An Introduction to Quantum Computing.
4. Jack D. Hidary, Quantum Computing: An Applied Approach.
5. Rodney Van Meter, Quantum Networking.
6. Daniel A. Lidar and Todd A. Brun, Quantum Error Correction.
7. Eric Ries, The Lean Startup.

Semester	MCA26305B: MOBILE COMPUTING WITH ANDROID	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	To introduce the concept of mobile android
LO2	To introduce the concept of different views of android
LO3	To understand the design aspects of android mobiles.
LO4	To make them familiar with SMS, email, service, binding and deploying APks.

COURSE OUTCOMES (CO):

Students successfully completing this module will be able to:

CO1	Familiarized with mobile android Terminology.
CO2	Understand and building interfaces.
CO3	Understand and creating menus
CO4	Gain knowledge about publishing, deploying APK files and Eclipse.

Course Content:

UNIT - I

What is Android? Features of Android, Architecture of Android, Eclipse, Android SDK, ADT, Creating Android virtual devices, Creating Application and Anatomy application.

Understanding Activities - Applying styles and themes to activity, hiding the activity title, displaying a dialog window, displaying a progress dialog. Linking Activities using intents. Calling built-in applications using intents.

UNIT - II

Understand the components of a screen, Adapting to display orientation, managing changes to screen orientation, creating the user interface programmatically, listening for UI notifications.

Basic views, pickers views, list views. Using images views to display pictures, using menus with views and some additional views.

UNIT - III

User preferences, persisting data to files, creating and using databases, sharing data in android, using a content provider, creating your own content provider, SMS messaging, e-mails and networking.

UNIT - IV

Creating own services, communicating between a service and an activity, binding activities to services, publishing, deploying APF files and eclipse.

PRESCRIBE TEXTBOOK

Beginning Android 4 Application Development, Wei-Meng Lee, Wiley

REFERENCE BOOKS

Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox).

Semester	MCA26305C: MICROSOFT DYNAMICS	L	T	P	C
III		3	1	0	4

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

LO1	Explore Dynamics 365 applications.
LO2	Discover business value in Dynamics 365 applications and how they support digital transformation.
LO3	Gain in-depth knowledge of CRM application modules.
LO4	Understand Dynamics 365 customization, configuration, integration, deployment types
LO5	Design and configure business units

COURSE OUTCOMES (CO):

At the end of the course, the student will be able to:

CO1	Understand Dynamics 365 framework
CO2	Examine the Design and configuration of Business Units, Configure Security Roles
CO3	Experiment with Creating and managing entities
CO4	Illustrate the Customization of forms, views and visualizations
CO5	Develop various process flows and solution management.

Course Content:

UNIT - I

Prerequisites

a) Install VS Code - https://code.visualstudio.com/download

b) Install Docker Engine - https://docs.docker.com/engine/install/

c) Install Al Visx file

I. Go to visual Studio code and then marketplace

II. Search AL and install the AL Language vsix file

d) Install Bc Container Helper - https://github.com/microsoft/navcontainerhelper/blob/master/NavContainerHelper.md

#Get Started Helper

e) Install Bc Container using BC Container helper - https://github.com/microsoft/navcontainerhelper/blob/master/NavContainerHelper.md

#Get Started Run

UNIT - II

Business Central – Basic Functional Knowledge

A) Getting Started - https://learn.microsoft.com/en-us/training/paths/get-started- dynamics-365- business-central/

B) Sales - https://learn.microsoft.com/en-us/dynamics365/business-central/sales-manage-sales

C) Purchase - https://learn.microsoft.com/en-us/dynamics365/business-central/purchasing-manage-purchasing

D) Inventory - https://learn.microsoft.com/en-us/dynamics365/business-central/inventory-manage-inventory

UNIT - III

Business Central–AL Development

a) Getting Started - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-Itpro/developer/devenv-get-started
b) Creating a Sample Project - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-extension-example
c) Keyboard Shortcuts - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-Itpro/developer/devenv-extension-example

d) Business Objects

I. Tables - https://learn.microsoft.com/en-us/dynamics365/business- central/dev-it pro/developer/dev env-tables-overview

II. Pages - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-pages-overview

III. Events - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-events-in-al

IV. Reports - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-reports

UNIT - IV

Business Central – AL Programming

a) Overview - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-Itpro/developer/devenv-reference-overview
b) DataTypes & Methods - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/methods-auto/library
c) Properties - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/properties/devenv-properties
d) Triggers - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/triggers-auto/devenv-triggers
e) AL Methods - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-Itpro/developer/devenv-al-methods
f) AL Simple Statements - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-al-simple-statements
g) AL Control Statements - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-al-control-statements

Semester	MCA26306: DATA MINING AND BIG DATA LAB	L	T	P	C
III		0	0	6	3

CYCLE – 1 Data Mining

(Using Python, Java, WEKA or any open-source data mining tool)

b. Write a program to Generate Association rules by using Apriori algorithm

c. Write a program to implement naïve Bayesian classification

d. Write a program to implement k-means clustering algorithm

e. Write a program to implement k-mediods clustering algorithm

f. Write a program to implement dbscan algorithm

CYCLE – 2 Hadoop

1. Implement the following Data structures in Java a) Linked Lists b) Stacks c) Queues

d) Set e) Map

2. Study and configure hadoop for big data

3. Hadoop commands

4. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

5. Implement Matrix Multiplication with Hadoop Map Reduce

Semester	MCA26307: CRYPTOGRAPHY AND NETWORK SECURITY LAB	L	T	P	C
III		0	0	6	3

1. Write a Java program to perform encryption and decryption using the following algorithms:

a) Ceaser Cipher

b) Substitution Cipher

c) Hill Cipher

2. Write a Java program to implement the DES algorithm logic.
3. Write a Java program to implement RSA Algorithm.
4. Write a C/JAVA program to implement the Blowfish algorithm logic.
5. Write a C/JAVA program to implement the Rijndael algorithm logic.
6. Using Java Cryptography, encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool.
7. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
8. Write a program to implement digital signature.
9. Compute common secret key between client and server using Diffie-Hellman key exchange technique. Perform encryption and decryption of message using the shared secret key (Use simple XOR operation to encrypt and decrypt the message).
10. Implement DSS algorithm for signing and verification of messages between two parties (obtain H (M) using simple XOR method of hash computation on M).

Semester	MCA26308: TECHNICAL REPORT WRITING	L	T	P	C
III		0	4	0	2

COURSE DESCRIPTION

This course develops the ability to prepare clear, accurate, well-organized technical documents in computer applications. It covers the planning, drafting, revision, and presentation of reports based on software development, experiments, literature studies, and analytical work.

Students learn to document methods and results, use evidence appropriately, prepare effective tables and figures, and follow consistent citation and formatting conventions.

LEARNING OBJECTIVES (LO):

1. LO1: Develop an understanding of the purpose, audience, and structure of technical documents.
2. LO2: Strengthen the ability to organize and communicate technical information.
3. LO3: Enable students to describe methods, implementations, results, and limitations accurately.
4. LO4: Develop skills in literature use, citation, referencing, and academic integrity.
5. LO5: Provide practical experience in preparing and revising a complete technical report.

COURSE OUTCOMES (CO):

Upon successful completion of the course, students will be able to:

- **CO1:** Plan a technical document according to its purpose and audience.
- **CO2:** Write clear, coherent, and appropriately structured technical content.
- **CO3:** Present methods, data, results, tables, and figures accurately.
- **CO4:** Apply consistent citation and referencing practices.
- **CO5:** Prepare, review, and revise a complete technical report using appropriate tools.

Course Content:

UNIT I: PRINCIPLES OF TECHNICAL WRITING

Purpose and characteristics of technical writing; technical and non-technical audiences; clarity, precision, conciseness, objectivity, and consistency; types of technical documents; understanding the writing task; defining scope; organizing information; outlining; paragraph development; terminology and abbreviations; appropriate tone; common writing errors.

UNIT II: STRUCTURE AND ORGANIZATION OF REPORTS

Title page; abstract and executive summary; table of contents; lists of figures and tables; introduction; problem statement; objectives and scope; background and literature review; methodology; implementation; results; discussion; conclusions; recommendations; references and appendices. Selection of report structure according to document purpose; heading hierarchy; logical transitions and cross-references.

UNIT III: DOCUMENTATION OF COMPUTING WORK

Writing software project reports; documenting requirements, system architecture, design decisions, algorithms, data sources, implementation, testing, and evaluation; describing experimental procedures; reporting datasets and preprocessing; specifying tools and relevant configurations; explaining reproducibility; presenting results and limitations; distinguishing observations from interpretations; preparing user and installation documentation.

UNIT IV: EVIDENCE, VISUAL PRESENTATION, AND REFERENCING

Selecting and evaluating information sources; integrating literature into a report; summarizing, paraphrasing, and quotation; avoiding plagiarism; applying a consistent reference style, such as IEEE, as prescribed by the Department; reference-management tools; preparing tables, charts, diagrams, equations, and code extracts; numbering and captions; labelling axes and units; acknowledging reproduced material; responsible use and verification of AI-assisted writing.

UNIT V: EDITING, REVIEW, AND DOCUMENT PRODUCTION

Drafting and revision; reviewing technical accuracy and completeness; peer review; proofreading; consistency of terminology, headings, numbering, and references; document preparation using word-processing or typesetting tools; styles and templates; automatic contents and cross-references; page layout; accessibility and readability; version management; final submission checks; converting a report into a concise presentation.

SUGGESTED PRACTICAL ACTIVITIES

1. Revise an unclear technical passage for clarity and accuracy.
2. Prepare an outline and problem statement for a computing project.
3. Write an abstract and an introduction.
4. Prepare a short literature review using credible sources.
5. Document an algorithm, software module, or experimental procedure.
6. Present a dataset or result using a properly labelled table and figure.
7. Prepare citations and a reference list in the prescribed style.
8. Conduct a peer review and revise the report in response to feedback.
9. Submit a complete technical report based on an approved project, experiment, or literature study.

EXPECTED DELIVERABLES

A report outlines selected draft sections, a peer-review record, and a final technical report. The final report should contain sufficient detail for a reader to understand the problem, approach, evidence, conclusions, and limitations.

SUGGESTED ASSESSMENT

Evaluation may consider organization, clarity, technical accuracy, quality of evidence, description of methods, presentation of results, appropriate use of tables and figures, citation accuracy, formatting consistency, and improvement through revision.

LEARNING RESOURCES

Departmental report templates, approved sample technical reports, official documentation for the tools used, the prescribed referencing guide, and relevant scholarly literature.



Semester	MCA26309: SWAYAM PLUS COURSES	L	T	P	C
III		0	0	0	2

*__NOTE:__ The credits assigned for SWAYAM PLUS Courses (MOOCs Mandatory Course) will not be counted for the GPA.

COURSE DESCRIPTION

This course component develops practical and employment-oriented skills through approved SWAYAM Plus courses. Preference shall be given to offerings related to Artificial Intelligence, Machine Learning, Data Science, Data Analytics, and associated software tools.

Students shall undertake a course that provides suitable practical learning through exercises, case studies, projects, or tool-based activities. The emphasis is on applying knowledge to clearly defined problems and demonstrating the skills acquired.

LEARNING OBJECTIVES (LO):

The course is designed to meet the objectives of:

1. LO1: Develop practical skills relevant to contemporary computing roles.
2. LO2: Strengthen the application of AI, ML, and Data Science tools.
3. LO3: Introduce students to professional workflows and problem-solving practices.
4. LO4: Encourage the creation of demonstrable work and learning portfolios.
5. LO5: Promote responsible, continuous, and employment-oriented learning.

COURSE OUTCOMES (CO):

Upon successful completion of the approved course, students will be able to:

- **CO1:** Use the tools and techniques introduced in the selected course.
- **CO2:** Complete relevant practical exercises or case studies.
- **CO3:** Apply acquired skills to a defined computing or data-analysis problem.
- **CO4:** Evaluate the quality, limitations, and responsible use of the resulting solution.
- **CO5:** Demonstrate learning through a portfolio, report, presentation, or viva voce.

RECOMMENDED AREAS FOR COURSE SELECTION

1. **Applied AI and ML:** Implementing and evaluating machine learning solutions using appropriate programming tools.
2. **Data Analytics and Visualization:** Preparing datasets, analyzing information, creating visualizations, and communicating findings.
3. **Python for AI and Data Science:** Practical programming, data processing, and use of analytical libraries.
4. **Generative AI Applications:** Effective use and evaluation of generative tools, output verification, and responsible application development.
5. **Applied Deep Learning:** Introductory practical work in image, text, or other suitable data domains.
6. **Deployment and Data Workflows:** Introductory model deployment, cloud-based analytical workflows, or related practices appropriate to the student's level.

The detailed instructional content shall be determined by the approved course. The Department should favor offerings with clear learning outcomes, suitable practical content, and verifiable assessment or completion requirements.

COURSE SELECTION AND APPROVAL

- Students shall select a course with prior departmental approval.
- The selected course should match the student's prerequisites and the intended learning outcomes of the MCA programme.
- Course workload, provider, assessment arrangements, certificate requirements, software access, and any applicable fees shall be reviewed before approval.
- Where both SWAYAM and SWAYAM Plus components occur in the programme, students should choose complementary courses.
- The same learning activity or certificate shall not be counted twice.

SUGGESTED PRACTICAL WORK

Depending on the selected course, students may undertake one of the following:

- Prepare and analyze a dataset using Python.
- Create an analytical dashboard and explain its findings.
- Build and evaluate a classification or regression model.
- Compare alternative machine learning approaches.
- Develop a small AI-enabled application.
- Demonstrate a text-analysis or image-analysis workflow.
- Document a practical case study using relevant tools.

Where predictive modelling is involved, students should use an appropriate evaluation method and avoid data leakage. Work involving personal or sensitive information shall follow applicable institutional requirements.

EXPECTED DELIVERABLES

- Evidence of enrolment and successful completion
- A brief description of the skills acquired
- Practical exercises, notebooks, source code, dashboards, or other relevant artefacts
- A short application report explaining the problem, method, results, and limitations
- A demonstration or viva voce as prescribed by the Department

Where the selected course does not include a practical submission itself, the Department may prescribe a proportionate application exercise aligned with its learning outcomes.

ASSESSMENT AND RECOGNITION

Evaluation should consider successful completion of the approved course and the student's ability to demonstrate the skills acquired. Any departmental assessment, grade conversion, or credit recognition shall follow the University's approved scheme. A provider certificate shall not automatically establish equivalence to an MCA credit-bearing course.

RECOMMENDED COURSES AND LEARNING RESOURCES

Students are encouraged to select practical, employment-oriented courses in Artificial Intelligence, Machine Learning, Data Science, Data Analytics, and associated software tools. The following courses are indicative options, subject to availability and prior departmental approval:

1. **AI / ML using Python**
   Recommended for developing practical skills in Python, data visualization, and building machine learning models.
   Course link: https://swayamplus.education.gov.in/courses/ai-ml-using-python
2. **Machine Learning Basics: Introduction to Machine Learning**
   Recommended for developing an introductory understanding of machine learning and its applications. The Department shall review the course depth and practical activities to confirm suitability for the student's level.
   Course link: https://swayamplus.education.gov.in/courses/machine-learning-basics-introduction-to-machine-learning
3. **SWAYAM Plus Portal**
   Students may identify other suitable courses in applied AI, machine learning, data analytics, Python programming, generative AI, and related areas. Preference should be given to courses with practical exercises, case studies, projects, or demonstrable skill outcomes.
   Portal link: https://swayamplus.education.gov.in/

SELECTION AND COMPLETION GUIDELINES

The Department shall verify course availability, prerequisites, workload, practical content, assessment arrangements, certification requirements, software access, and any applicable fees before approving enrolment. Where a listed course is unavailable, an equivalent course in an approved subject area may be selected.

Materials prescribed by the selected course, relevant official software documentation, datasets, and faculty-recommended resources shall support learning. Students shall submit evidence of successful completion and demonstrate the acquired skills through practical work, a report, presentation, or viva voce, as prescribed by the Department.

Where both SWAYAM and SWAYAM Plus components occur in the programme, students should select complementary courses. The same course or certificate shall not be counted twice. Any credit recognition shall follow the University's approved scheme.

SEMESTER IV

Semester	MCA26401: PROJECT WORK	L	T	P	C
IV		0	0	36	12

Internal Assessment: 100 Marks

Assessment Pattern: Four Internal Reviews × 25 Marks

External Assessment: 200 Marks

COURSE DESCRIPTION

Project Work enables students to integrate the knowledge and skills acquired during the MCA programme to investigate and solve a clearly defined computing problem. Students shall work in batches under the supervision of an allotted faculty guide and undertake problem identification, literature review, requirements analysis, design, implementation, testing, evaluation, and documentation.

The Department shall conduct four formal reviews at defined stages to assess project development, individual contributions, and improvement based on feedback. The completed project shall demonstrate technical understanding, systematic development, appropriate evaluation, and professional reporting.

LEARNING OBJECTIVES (LO):

The Project Work is designed to meet the objectives of:

1. LO1: Enable students to apply computing knowledge to a practical or research-oriented problem.
2. LO2: Develop skills in literature review, problem formulation, system design, implementation, and evaluation.
3. LO3: Promote teamwork, project planning, accountability, and professional communication.
4. LO4: Encourage the use of relevant research literature, applicable standards, and recognized development practices.
5. L05: Develop the ability to improve a solution through regular review, testing, and feedback.

COURSE OUTCOMES (CO):

Upon successful completion of Project Work, students will be able to:

- **CO1:** Identify a relevant problem and formulate achievable objectives supported by literature or stakeholder requirements.
- **CO2:** Design and implement an appropriate solution using suitable technologies and development practices.
- **CO3:** Test and evaluate the solution against defined requirements and measurable criteria.
- **CO4:** Work effectively in a batch and demonstrate individual technical contributions.
- **CO5:** Prepare a structured project report and demonstrate the completed work through presentation and viva-voce.

1. BATCH FORMATION AND ALLOTMENT OF FACULTY GUIDES

1. Students shall form project batches of a size prescribed by the Department, considering the scope of work, student strength, and available facilities.
2. The Head of the Department or Project Coordinator shall approve batch formation and allot a faculty guide to each batch.
3. Each faculty guide should supervise an allotted set of students or batches. The distribution shall consider faculty expertise and a reasonable supervisory workload.

4. Each batch shall submit the names and registration numbers of its members, proposed topic, allotted guide, and preliminary allocation of responsibilities.
5. Every member shall participate in technical work, documentation, review presentations, and the final demonstration.
6. Changes in batch composition, project topic, or faculty guide shall require departmental approval.

2. RESPONSIBILITIES OF THE FACULTY GUIDE

The faculty guide shall:

1. Assist students in selecting a suitable problem and defining its objectives and scope.
2. Guide the literature review, selection of methods, system design, implementation, and evaluation.
3. Help the batch prepare a realistic schedule with identifiable milestones and individual responsibilities.
4. Conduct regular supervisory meetings and verify progress through demonstrations, code, experiments, reports, or other relevant evidence.
5. Review the project diary and provide specific feedback and corrective actions.
6. Monitor the contribution and understanding of each student.
7. Verify that review comments are addressed and that the work follows appropriate academic and professional practices.
8. Review the final report and advise the batch on readiness for submission.

3. PROJECT SELECTION AND APPROVAL

Projects may be application-oriented, research-oriented, industry-sponsored, or directed toward a societal or institutional need. Suitable areas include Artificial Intelligence, Machine Learning, Data Science, Software Engineering, Cybersecurity, Cloud Computing, Internet of Things, Database Systems, Web and Mobile Applications, and related computing domains.

The Department shall approve projects based on the following criteria:

- **Relevance:** The project addresses a clearly identified problem appropriate to the MCA programme.
- **Technical depth:** The work requires meaningful analysis, design, implementation, and evaluation.
- **Feasibility:** The objectives are achievable within the available time, resources, and student capabilities.
- **Evidence base:** The proposal is supported by relevant literature, stakeholder needs, or documented limitations of existing systems.
- **Contribution:** The proposed work offers an identifiable implementation, adaptation, integration, comparative evaluation, or improvement.
- **Evaluability:** Success can be assessed using defined functional requirements, test cases, performance measures, or other suitable criteria.

RESEARCH REFERENCES AND STANDARDS

1. Students should consult relevant peer-reviewed publications, including IEEE journals and conference proceedings, and other credible academic or technical sources.
2. Where appropriate, the batch shall identify one or more base papers and explain the proposed implementation, extension, comparison, or improvement.
3. A project based on an existing paper shall clearly distinguish the original authors' work from the students' contribution.

4. Students shall identify relevant IEEE, ISO/IEC, or other recognized standards where applicable to the project. The report shall state the specific standard and the aspects of the work to which it applies.
5. Citing an IEEE paper alone shall not be treated as evidence that a project complies with an IEEE standard.
6. Projects shall follow appropriate practices for requirements documentation, design, coding, testing, version control, and reporting.
7. An application addressing a documented practical need may be approved without a single base paper, provided it demonstrates sufficient technical depth and evaluation.

4. PROJECT PROPOSAL

Each batch shall submit a proposal containing:

1. Project title and batch details.
2. Background and problem statement.
3. Literature review or study of existing systems.
4. Identified limitations, needs, or research gap.
5. Objectives and scope.
6. Proposed methodology and system overview.
7. Required software, hardware, datasets, and other resources.
8. Expected deliverables and evaluation criteria.
9. Work schedule and allocation of individual responsibilities.
10. References and applicable standards were relevant.

5. SUPERVISION AND PROGRESS MONITORING

1. Each batch shall maintain a project diary recording meetings, tasks completed, individual contributions, difficulties, feedback, and planned actions.
2. The faculty guide shall review the diary regularly.
3. Students shall maintain dated versions of source code, design documents, experimental results, and reports. A version-control repository should be used where appropriate.
4. The Department shall announce the four review dates and expected deliverables at the beginning of the project period.
5. A Project Review Committee constituted by the Head of the Department shall conduct the reviews. The committee shall include the faculty guide and other nominated faculty members.
6. Each review shall include a presentation, examination of relevant work, and individual questioning.
7. The committee shall record marks, observations, corrective actions, and milestones for the next review.
8. At subsequent reviews, students shall demonstrate how they have addressed earlier feedback.

6. INTERNAL ASSESSMENT SCHEME

The 100 internal marks shall be awarded through four reviews of 25 marks each.

Review I: Problem Identification and Project Planning — 25 Marks

Expected Stage: Initial stage of the project.

Required Deliverables: Proposal, literature review or existing-system study, objectives, feasibility assessment, work plan, and individual responsibility statement.

Assessment Criterion	Marks
Relevance and clarity of the problem statement and objectives	5
Literature review or existing-system study and identification of the gap or need	5
Feasibility, proposed methodology, and evaluation plan	5
Work schedule, milestones, and allocation of responsibilities	5
Individual understanding, contribution, and response to preliminary guide feedback	5
Total	**25**

Review II: Requirements, Design, and Initial Prototype — 25 Marks

Expected Stage: Completion of detailed analysis and design, with an initial prototype or preliminary experiments.

Required Deliverables: Requirements document, architecture or workflow, detailed design, prototype or experimental setup, updated diary, and response to Review I comment.

Assessment Criterion	Marks
Completeness and clarity of requirements or experimental specifications	5
Quality of architecture, methodology, and detailed design	5
Initial prototype, proof of concept, or preliminary experimental work	5
Demonstrated progress and action taken on Review I feedback	5
Individual contribution, technical understanding, and review presentation	5
Total	**25**

Review III: Implementation and Preliminary Evaluation — 25 Marks

Expected Stage: Substantial completion of implementation and initial testing or evaluation.

Required Deliverables: Working modules or experimental pipeline, integration progress, preliminary results, test records, updated documentation, and response to Review II comments.

Assessment Criterion	Marks
Implementation of planned functionality or experimental methods	5
Integration, technical quality, and appropriate development practices	5
Preliminary testing, evaluation, and analysis of results	5
Demonstrated improvement and action taken on Review II feedback	5
Individual contribution, problem-solving ability, and technical explanation	5

Total	**25**

Review IV: Final Demonstration and Project Completion — 25 Marks

Expected Stage: Completion of the project before final submission.

Required Deliverables: Completed system or experimental work, final evaluation, project report, source code and supporting materials, user or execution instructions, and response to Review III comments.

Assessment Criterion	**Marks**
Final demonstration and achievement of approved objectives	5
Testing, evaluation, analysis of results, and recognition of limitations	5
Quality and completeness of the report and supporting documentation	5
Demonstrated improvement and action taken on Review III feedback	5
Individual contribution, presentation, and viva voce	5
Total	**25**

Total Internal Marks = Review I + Review II + Review III + Review IV = 100 Marks

7. PRINCIPLES FOR AWARDING MARKS

1. Marks shall be awarded to each student individually, supported by the quality of the batch's work and evidence of the student's contribution and understanding.
2. Membership in the same batch shall not automatically result in identical marks.
3. Assessment shall consider achievement of milestones, technical quality, problem-solving, and improvement following feedback.
4. Improvement shall be demonstrated through evidence such as corrected designs, completed functionality, resolved defects, stronger evaluation, or clearer documentation.
5. Marks shall not be awarded solely for attendance, presentation quality, the number of pages in the report, or unsupported claims of completion.
6. Previous review marks remain recorded. Later improvements shall be recognized in subsequent reviews.
7. Students who have already met an expected standard may earn full progress marks by maintaining quality and completing the next agreed milestones.
8. Missed reviews and any permitted reassessment shall be handled according to departmental procedures and University regulations.

8. TESTING AND EVALUATION REQUIREMENTS

Each batch shall evaluate its project against the objectives and criteria approved during the initial reviews.

- Software applications shall include relevant functional, integration, usability, performance, or security testing, as appropriate.
- AI and ML projects shall use suitable datasets, appropriate separation of training and evaluation data, relevant metrics, and meaningful comparisons.
- Research-oriented projects shall explain the experimental procedure, assumptions, results, and limitations.

- Claimed improvements shall be supported by reproducible tests, measurements, comparisons, or documented user evaluation.
- Unachieved objectives and unresolved limitations shall be reported clearly.

9. PROJECT REPORT FORMAT

The final report shall follow the departmental template and ordinarily include:

1. Title page.
2. Certificate from the faculty guide and Department.
3. Student declaration.
4. Acknowledgements.
5. Abstract.
6. Table of contents.
7. Lists of figures, tables, and abbreviations, where applicable.
8. Introduction, problem statement, objectives, and scope.
9. Literature review or study of existing systems.
10. Requirements analysis and proposed methodology.
11. System architecture and detailed design.
12. Implementation.
13. Testing, results, and discussion.
14. Conclusions, limitations, and future scope.
15. References in a consistent format, preferably IEEE referencing style.
16. Appendices, including relevant instructions, sample outputs, and individual contribution statements.

The structure may be adapted with the guide's approval to suit experimental, data-science, or research-oriented projects.

10. FINAL SUBMISSION

Each batch shall submit:

- The completed project report.
- Source code and relevant configuration files.
- Dataset references and permitted supporting data, where applicable.
- Installation, execution, and user instructions.
- Test cases, evaluation records, and results.
- Presentation slides.
- Project diary and review-action records.
- A statement of each member's contribution.

Each student shall participate in the final demonstration and explain their own work as well as the overall project.

11. ACADEMIC INTEGRITY AND PROFESSIONAL PRACTICE

All borrowed ideas, code, figures, datasets, and text shall be acknowledged. Students shall identify reused components and explain their own contribution. Fabricated results purchased projects presented as original work, and unauthorized copying are not acceptable.

AI-assisted tools may be used in accordance with University policy. Students remain responsible for the accuracy, originality, and technical validity of the submitted work and shall be able to explain and demonstrate it independently.

Research publication, patent filing, or participation in competitions may be encouraged where appropriate, but shall not substitute for the required project work, reviews, and assessment.