UNIT IV

Knowledge Representation: Introduction, Graphical Sketches and the Human Window, Graphs and the Bridges of Königsberg Problem, Search Trees, Representational Choices, Production Systems, Object Orientation, Frames, Semantic Networks

UNIT V

Production Systems: Introduction, Background, Production Systems and Inference Methods, Production Systems and Cellular Automata, Stochastic Processes and Markov Chains, Basic Features and Examples of Expert Systems

Prescribed Textbook

1. Stephen Lucci, Danny Kopec, Artificial Intelligence in the 21st Century-A Living Introduction, Mercury Learning and Information, Second Edition, 2016

Reference Books

1. Russell, Norvig: Artificial Intelligence, A Modern Approach, Pearson Education, Second Edition, 2004
2. Rich, Knight, Nair: Artificial Intelligence, Tata McGraw Hill, Third Edition, 2009
3. Saroj Kaushik. Artificial Intelligence. Cengage Learning, 2011

SEMESTER-III

M.Sc. II Year – III Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MS26301	PC	Advance Java Programming	30	70	3	1	0	4
2	MS26302	PC	Machine Learning	30	70	3	1	0	4
3	MS26303	PC	R Programming	30	70	3	1	0	4
4	MS26304A	PE-III	1. Operating System (OS)	30	70	3	1	0	4
	MS26304B		2. Cyber Law						
	MS26304C		3. Cryptography & Network Security						
5	MS26305A	PE-IV	1. WISER – Quantum Foundation and Applied Technologies	30	70	3	1	0	4
	MS26305B		2. Neural Networks & Deep Learning						
	MS26305C		3. Internet of Things (IoT)						
	MS26305D		4. Blockchain Technology						
6	MS26306	PC-Lab	Advance Java Programming Lab	25	50	0	0	6	3
7	MS26307	PC-Lab	Machine Learning Lab	25	50	0	0	6	3
8	MS26308	PC-Lab	R Programming Lab	50	-	0	0	4	2
9	MS26309	MC	SWAYAM Plus Courses - An initiative by Ministry of Education (Govt. of India)	-	-	0	0	0	2*
			Total	**250**	**450**	**15**	**5**	**16**	**28**

PC – Program Course

PE – Professional Elective

HS – Humanities and Social Sciences

MC – Mandatory Courses (MOOCs)

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

Advanced Java Programming					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26301	**PC**	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of:

LO1	Design and develop Web applications
LO2	Create web pages using HTML, DHTML and Cascading Styles sheets.
LO3	Analyze and build interactive web applications using JSP and Servlets.
LO4	Design and develop Web applications
LO5	Designing Enterprise based applications by encapsulating an application's business logic

Course Outcomes (CO):

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

CO1	Explain the technologies used in web applications.
CO2	Demonstrate HTML5, CSS, JavaScript coding for web applications
CO3	Design creative websites using object based scripting concepts
CO4	Learn to access database through Java programs, using Java Data Base Connectivity (JDBC)
CO5	Create dynamic web pages, using Servlets and JSP

UNIT I

HTML Common Tags: List, Tables, images, forms, Frames, Cascading Style Sheets.

Java Script: Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script.

UNIT II

XML - Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors DOM and SAX

CGI Scripting- What is CGI? – Developing CGI applications – Processing CGI – Returning a Basic HTML page – Introduction to CGI.pm – CGI.pm methods – Creating HTML pages dynamically.

UNIT III

JDBC Introduction to JDBC – Connections – Internal Database Connections – Statements – Results Sets

- Prepared Statements - Callable Statements.

Network Programming and RMI why networked Java – Basic Network Concepts – looking up Internet Addresses – URLs and URIs – UDP Datagram's and Sockets – Remote Method Invocation.

UNIT –IV

Web Servers, Tomcat web server, Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat, Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat.

Servlets

Introduction to Servlets Lifecycle of a Servlet, JSDK, The Servlet API, The javax.servlet Package, Reading Servlet parameters, Reading Initialization parameters. The javax.servlet HTTP package,Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues.

UNIT-V

Introduction to JSP The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment

Prescribed Textbooks

1. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech (units I, II)
2. Java Programming with JDBC ;Donald Bales, O'Reilly (Unit III)
3. Java Network Programming, elliotte Rusty Harold, 3rd Edition, O'Reilly (Unit III)
4. Java Server Pages – Hans Bergsten, SPD O'Reilly (Unit IV)

Reference Textbooks

1. Robert W. Sebesta, "Programming the World Wide Web", Third Edition, Pearson Education (2007).
2. Anders Moller and Michael schwartzbach, "An Introduction to XML and Web Technologies", Addison Wesley (2006)
3. Jeffrey C. Jackson, "Web Technologies – A Computer Science Perspective", Pearson Education (2008).
4. H. M. Deitel, P. J. Deitel, "Java How to Program", Sixth Edition, Pearson Education (2007).

Machine Learning					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
MS26302	**PC**	**L**	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
		3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of:

LO1	To introduce to the students the basic concepts and fundamentals of machine learning
LO2	To develop skills of implementing machine learning techniques
LO3	To familiarize the students with latest technologies
LO4	To implement machine learning solutions to classification, regression
LO5	To implement machine learning algorithms for clustering

Course Outcomes (CO):

Students successfully completing this module will be able to:

CO1	How to make a computer program learn from experience
CO2	Illustrate the significance of concept learning
CO3	Representation of decisions and decision making explicitly
CO4	Construct finite and infinite Hypothesis spaces for computational learning
CO5	Apply Inductive and Analytical learning in developing learning tasks

UNIT – I

Introduction - Well-posed learning problems, Designing a learning system, Perspectives and issues in machine learning

Concept learning and the General to Specific Ordering – Introduction, A concept learning task, Concept learning as search, Find-S finding a maximally specific hypothesis, Version spaces and the Candidate-Elimination algorithm, Remarks on version spaces and Candidate-Elimination, Inductive Bias

UNIT – II

Decision Tree learning – Introduction, Decision tree representation, Appropriate problems for decision tree learning, The basic decision tree learning algorithm, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning

Evaluation Hypotheses – Motivation, Estimation hypothesis accuracy, Basics of sampling theory, A general approach for deriving confidence intervals, Difference in error of two hypotheses, Comparing learning algorithms

UNIT-III

Bayesian learning – Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum likelihood and least squared error hypotheses, Maximum likelihood hypotheses for predicting probabilities, Minimum description length principle, Bayes optimal classifier, Gibbs algorithm, Naïve Bayes classifier, An example learning to classify text, Bayesian belief networks The EM algorithm

UNIT-IV

Computational learning theory – Introduction, Probability Learning an Approximately Correct Hypothesis, Sample Complexity for Finite Hypothesis Space, Sample Complexity for infinite Hypothesis Spaces, The Mistake Bound Model of Learning

Instance-Based Learning- Introduction, k -Nearest Neighbour Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning

UNIT- V

Genetic Algorithms – Motivation, Genetic Algorithms, An illustrative Example, Hypothesis Space Search, Genetic Programming, Models of Evolution and Learning, Parallelizing Genetic Algorithms

Combining Inductive and Analytical Learning – Motivation, Inductive-Analytical Approaches to Learning, Using Prior Knowledge to Initialize the Hypothesis, Using Prior Knowledge to Alter the Search Objective, Using Prior Knowledge to Augment Search Operators

Reinforcement Learning – Introduction, The Learning Task, Q Learning, Non-Deterministic, Rewards and Actions, Temporal Difference Learning, Generalizing from Examples, Relationship to Dynamic Programming

Prescribed Textbooks

- Machine Learning – Tom M. Mitchell, - MGH

Reference Books

- Introduction to Machine Learning,- Ethem Alpaydin, - PHI
- Machine Learning An Algorithmic Perspective, Stephen Marsland, Taylor & Francis

R Programming					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26303	**PC**	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of:

LO1	The basics of statistical computing and data analysis
LO2	How to implement data structure in R
LO3	R loop functions and debugging tools
LO4	Lists and Vectors concepts in R
LO5	Factors and levels in R

Course Outcomes (CO):

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

CO1	Explain critical R programming concepts
CO2	Demonstrate how to install and configure RStudio
CO3	Apply OOP concepts in R programming
CO4	Explain the use of data structure and loop functions
CO5	Apply various concepts to write programs in R

UNIT I

Introduction : Overview of R, R data types and objects, reading and writing data, sub setting R Objects, Essentials of the R Language, Installing R, Running R, Packages in R, Calculations, Complex numbers in R, Rounding, Arithmetic, Modulo and integer quotients, Variable names and assignment, Operators, Integers, Factors, Logical operations

UNIT II

Control structures, functions, scoping rules, dates and times, Introduction to Functions, preview of Some Important R Data Structures, Vectors, Character Strings, Matrices, Lists, Data Frames, Classes.

UNIT III

Vectors: Generating sequences, Vectors and subscripts, Extracting elements of a vector using subscripts, Working with logical subscripts, Scalars, Vectors, Arrays, and Matrices, Adding and Deleting Vector Elements, Obtaining the Length of a Vector, Matrices and Arrays as Vectors Vector Arithmetic and Logical Operations, Vector Indexing, Common Vector Operations

UNIT IV

Lists: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List, Extended Example: Text Concordance Accessing List Components and Values Applying Functions to Lists, Data Frames, Creating Data Frames, Accessing Data Frames, Other Matrix- Like Operations

UNIT V

Factors and Levels, Common Functions Used with Factors, Working with Tables, Matrix/Array-Like Operations on Tables , Extracting a Sub table, Finding the Largest Cells in a Table, Math Functions, Calculating a Probability, Cumulative Sums and Products, Minima and Maxima, Calculus, Functions for Statistical Distributions

Prescribed Textbooks:

1. Roger D. Peng, “R Programming for Data Science”, 2012
2. Norman Matloff, “The Art of R Programming- A Tour of Statistical Software Design”, 2011

Reference Textbooks:

1. Jared P. Lander, "R for Everyone - advanced analytics and Graphics", 2nd Edition, Addison-Wesley.
2. Paul Teetor, "R Cookbook", 2nd Edition, O'Reilly publications.
3. Garrett Grolemund, Hadley Wickham, “Hands-On Programming with R: Write Your Own Functions and Simulations”, 1st Edition, 2014

Operating Systems					M.Sc. 2nd year 3rd semester			
Code	Category	Hours / Week			Credits	MARKS		
		L	T	P	C	Internal	End Exam	Total
MS26304A	PE – III	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of appreciate

1. LO1: Understanding the role of an operating system
2. LO2: making aware of the issues in management of resources like processor, memory and input-output
3. LO3: Understanding file management techniques.

Course Outcomes (CO):

Students successfully completing this module will be able to

1. CO1: Understands what an operating system and role it is playing.
2. CO2: Get high level understanding of the structure of operating systems, applications, and the relationship between them
3. CO3: Gather knowledge of the services provided by operating systems
4. CO4: Get exposure to some details of major OS concepts.

UNIT–I

Introduction: What Operating Systems Do – Computer System Organization – Computer system Architecture – Operating System Structure – Operating System Operations – Process Management – Memory Management – Storage Management – Protection and Security – Distributed Systems – Special purpose Systems – Computing Environments.

System Structure: Operating System Services – User Operating System Interface – System Calls – Types of System Calls – System Programs – Operating System Design and Implementation – Operating System Structure – Virtual Machine – Operating System Generation – System Boot.

Process Concept: Overview – Process Scheduling – Operations on Processes – Inter process Communication – Examples of IPC Systems – Communication in Client Server Systems.

UNIT–II

Multithreaded Programming: Overview – Multithreading Models – Thread Libraries – Threading Issues – Operating System Examples.

Process Scheduling: Basic Concepts – Scheduling Criteria – Scheduling Algorithms – Multiple Processor Scheduling – Thread Scheduling.

UNIT–III

Synchronization : Background – The Critical Section Problem – Peterson's solution – Synchronization Hardware – Semaphores – Classic Problem of Synchronization – Monitors – Synchronization Examples – Atomic Transaction.

Deadlocks: System Model – Deadlock Characterization – Methods for Handling Deadlocks – Deadlock Prevention – Deadlock Avoidance – Deadlock Detection – Recovery from Deadlock.

UNIT–IV

Memory Management Strategies: Background – Swapping – Contiguous Memory Allocation – Paging – Structure of the Page Table – Segmentation – Example The Intel Pentium.

Virtual Memory Management: Background – Demand Paging – Copy on Write – Page Replacement – Allocation of Frames – Thrashing.

UNIT–V

File System: File Concept – Access Methods – Directory Structure – File System Mounting – File Sharing – Protection.

Implementing File Systems: File System Structure – File System Implementation – Directory Implementation – Allocation Methods – Free Space Management – Efficiency and Performance –Recovery – Log structured File Systems.

Secondary Storage Structure: Overview of Mass Storage Structure – Disk Structure – Disk Attachment – Disk Scheduling – Disk Management – Swap Space Management – RAID structure.

I/O Systems: Overview – I/O Hardware – Application I/O Interface – Kernal I/O Interface – Transforming I/O requests to Hardware Operations – Streams – Performance.

Prescribed Textbook

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. "Operating System Principles", Seventh Edition, Wiley.

Chapters 1.1 – 1.12, 2.1 – 2.10, 3.1 – 3.6, 4.1 – 4.5, 5.1 – 5.5, 6.1 – 6.9 , 7.1 – 7.7 , 8.1 – 8.7, 9.1 – 9.6, 10.1 – 10.6, 11.1 – 11.8, 12.1 – 12.7, 13.1 – 13.7

Reference Books

1.William Stallings, "Operating Systems – Internals and Design Principles", Fifth Edition, Pearson Education (2007)

2.Achyut S Godbole, "Operating Systems", Second Edition, TMH (2007).

3.Flynn/McHoes, "Operating Systems", Cengage Learning (2008).

4. Deitel & Deitel, "Operating Systems", Third Edition, Pearson Education (2008).

Cyber Laws					M.Sc. 2nd year 3rd semester			
Code	Category	Hours / Week			Credits	MARKS		
		L	T	P	C	Internal	End Exam	Total
MS26304B	PE – III	3	1	0	4	30	70	100

Learning Objectives (LO):

The objective of this course is to:

1. LO1: Enable learners to understand, explore, and acquire a critical understanding of Cyber Law
2. LO2: Develop competencies for dealing with frauds and deceptions (confidence tricks, scams) and other cyber crimes for example, child pornography etc. that are taking place via the Internet.
3. LO3: Make learners conversant with the social and intellectual property issues emerging from 'Cyberspace'.
4. LO4: Explore the legal and policy developments in various countries to regulate Cyberspace;
5. LO5: Develop the understanding of relationship between commerce and cyberspace; and give learners in depth knowledge of Information Technology Act and legal framework of Right to Privacy, Data Security and Data Protection.

Course Outcomes (CO):

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

1. CO1: Critically evaluate ongoing developments in law relating to information technologies
2. CO2: Display an understanding of how these developments relate to one another.
3. CO3: Examine areas of doctrinal and political debate surrounding rules and theories;
4. CO4: Evaluate those rules and theories in terms of internal coherence and practical outcomes;
5. CO5: Draw on the analysis and evaluation contained in primary and secondary sources

UNIT I

Introduction: Computers and its Impact in Society, Overview of Computer and Web Technology, Need for Cyber Law, Cyber Jurisprudence at International and Indian Level.

UNIT II

Cyber Law- International Perspectives: UN &International Telecommunication Union (ITU)Initiatives, Council of Europe -Budapest Convention on Cybercrime, Asia-Pacific Economic Cooperation(APEC), Organization for Economic Co-operation and Development(OECD), World Bank, Commonwealth of Nations.

UNIT III

Constitutional& Human Rights Issues in Cyberspace: Freedom of Speech and Expression in Cyberspace, Right to Access Cyberspace – Access to Internet, Right to Privacy, Right to Data Protection.

UNIT IV

Cyber Crimes& Legal Framework: Cyber Crimes against Individuals, Institution and State, Hacking, Digital Forgery, Cyber Stalking/Harassment, Cyber Pornography, Identity Theft & Fraud, Cyber terrorism, Cyber Defamation, Different offences under IT Act, 2000.

UNIT V

Cyber Torts: Cyber Defamation, Different Types of Civil Wrong sunder the IT Act, 2000, Intellectual Property Issues in Cyber Space, Interface with Copyright Law, Interface with Patent Law, Trade marks & Domain Names Related issues

Reference Books

1. Chris Reed & John Angel, Computer Law, OUP, New York, (2007).
2. Justice Yatindra Singh, Cyber Laws, Universal Law Publishing Co., New Delhi, (2012).
3. Verma K, Mittal Raman, Legal Dimensions of Cyber Space, Indian Law Institute, New Delhi, (2004)
4. Jonthan Rosenoer, Cyber Law, Springer, New York, (1997).
5. Sudhir Naib, The Information Technology Act, 2005: A Handbook, OUP, New York, (2011)
7. S. R. Bhansali, Information Technology Act, 2000, University Book House Pvt. Ltd., Jaipur (2003).
8. Vasu Deva, Cyber Crimes and Law Enforcement, Commonwealth Publishers, New Delhi, (2003).

Cryptography & Network Security					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26304C	**PE – III**	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of:

LO1	To provide introduction to the concept of Network Security Model and Cryptography systems.
LO2	To give the knowledge of Digital Signature and other Security Measures available.
LO3	To familiarize with the various techniques like PGP and S/MIME.
LO4	To showcase IP Security Architecture & Transport Layer Security to identify the vulnerability of the Internet systems and recognize the mechanisms of the attacks
LO5	To explain the firewall design principles and various intrusion detection system.

Course Outcomes (CO):

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

CO1	Identify and classify particular examples of attacks,
CO2	Compare and contrast symmetric and asymmetric encryption systems and their vulnerability to attack, and explain the characteristics of hybrid systems,
CO3	Describe the use of hash functions and explain the characteristics of one-way and collision-free functions,
CO4	Describe and distinguish between different mechanisms to assure the freshness of a message,
CO5	Explain the role of third-party agents in the provision of authentication services,

UNIT I

Introduction – security Trends, OSI security Architecture, security attacks, security services, security mechanisms, A model for network security

Conventional Encryption Classical Techniques - Symmetric cipher model, substitution techniques, transposition techniques, rotor machines, steganography

Conventional Encryption Modern Techniques – Block cipher principles, DES, strength of DES, Differential and linear cryptanalysis, Block cipher design principles.

UNIT II

Confidentiality using Symmetric encryption – Placement of encryption function, traffic confidentiality, key distribution, random number generation.

Public-key cryptography & RSA - Principles of Public key crypto systems, RSA algorithm, Key management, Diffie Hellman key exchange, elliptic curve cryptography.

UNIT III

Message authentication and Hash functions - Authentication requirements, Authentication functions, Message Authentication codes, Hash functions, Security of MAC's and hash functions

Digital signatures and Authentication Protocols - Digital signatures, Authentication Protocols, DSS

UNIT IV

Electronic Mail Security - PGP, S/MIME

IP Security - IP Security Overview, IP Security architecture, Authentication header, encapsulating security payload, combining security associations, key management.

UNIT V

Fire Walls - Firewall design principles, trusted systems

Prescribed Textbook

William Stallings, "Cryptography & Network Security Principles and Practices", 4th Edition, Pearson Education

Chapters 1,2,3,7,8,9,10,11,13,15,16,20

Reference Book

Bruce Schneier - Applied Cryptography - Wiley - second edition Davies & Price Security for computer networks, Wilsey (1984).

Quantum Foundation And Applied Technologies					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26305A	**PE – IV**	3	1	0	4	30	70	100

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.

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 visualizing 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.

Neural Networks & Deep Learning					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26305B	**PE – IV**	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of:

- LO1: To understand the basics in deep neural networks
- LO2: To understand the basics of associative memory and unsupervised learning networks
- LO3: To apply CNN architectures of deep neural networks
- LO4: To analyze the key computations underlying deep learning, then use them to build and train
- LO5: Deep neural networks for various tasks. To apply autoencoders and generative models for suitable applications.

Course Outcomes (CO):

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

- **CO1:** Apply Convolution Neural Network for image processing.
- **CO2:** Understand the basics of associative memory and unsupervised learning networks.
- **CO3:** Apply CNN and its variants for suitable applications.
- **CO4:** Analyze the key computations underlying deep learning and use them to build and train deep neural networks for various tasks.
- **CO5:** Apply autoencoders and generative models for suitable applications.

UNIT I

INTRODUCTION: Neural Networks-Application Scope of Neural Networks-Artificial Neural Network: An Introduction Evolution of Neural Networks-Basic Models of Artificial Neural Network- Important Terminologies of ANNs-Supervised Learning Network.

UNIT II

ASSOCIATIVE MEMORY AND UNSUPERVISED LEARNING NETWORKS: Training Algorithms for Pattern Association-Auto associative Memory Network-Hetero associative Memory Network-Bidirectional Associative Memory (BAM)-Hopfield Networks-Iterative Auto associative Memory Networks-Temporal Associative Memory Network-Fixed Weight Competitive Nets- Kohonen Self-Organizing Feature Maps-Learning Vector Quantization- Counter propagation Networks-Adaptive Resonance Theory Network.

UNIT III

THIRD-GENERATION NEURAL NETWORKS: Spiking Neural Networks-Convolutional Neural Networks-Deep Learning Neural Networks-Extreme Learning Machine Model- Convolutional Neural Networks: The Convolution Operation – Motivation – Pooling – Variants of the basic Convolution Function – Structured Outputs – Data Types – Efficient Convolution Algorithms – Neuroscientific Basis – Applications: Computer Vision, Image Generation, Image Compression.

UNIT IV

DEEP FEEDFORWARD NETWORKS: History of Deep Learning- A Probabilistic Theory of Deep Learning- Gradient Learning – Chain Rule and Backpropagation - Regularization: Dataset Augmentation – Noise Robustness -Early Stopping, Bagging and Dropout - batch normalization- VC Dimension and Neural Nets.

UNIT V

RECURRENT NEURAL NETWORKS: Recurrent Neural Networks: Introduction – Recursive Neural Networks – Bidirectional RNNs – Deep Recurrent Networks – Applications: Image Generation, Image Compression, Natural Language Processing. Complete Auto encoder, Regularized Autoencoder, Stochastic Encoders and Decoders, Contractive Encoders.

Prescribed Textbooks:

- Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016.
- Francois Chollet, “Deep Learning with Python”, Second Edition, Manning Publications, 2021.

Reference Books:

- Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow”, Oreilly, 2018.
- Josh Patterson, Adam Gibson, “Deep Learning: A Practitioner’s Approach”, O’Reilly Media, 2017.
- Charu C. Aggarwal, “Neural Networks and Deep Learning: A Textbook”, Springer International Publishing, 1st Edition, 2018.
- Learn Keras for Deep Neural Networks, Jojo Moolayil, Apress,2018
- Deep Learning Projects Using TensorFlow 2, Vinita Silaparasetty, Apress, 2020
- Deep Learning with Python, FRANÇOIS CHOLLET, MANNING SHELTER ISLAND,2017. 7. S Rajasekaran, G A Vijayalakshmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithm, Synthesis and Applications”, PHI Learning, 2017.
- Pro Deep Learning with TensorFlow, Santanu Pattanayak, Apress, 2017
- James A Freeman, David M S Kapura, “Neural Networks Algorithms, Applications, and Programming Techniques”, Addison Wesley, 2003.

Internet of Things					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26305C	**PE – IV**	3	1	0	4	30	70	100

Learning Objectives (LO):

The course is designed to meet the objectives of

- LO1: To introduce the concept of IoT
- LO2: To introduce the concept of M2M
- LO3: To understand the logical design
- LO4: To make them familiar with IoT devices, endpoints and designing

Course Outcomes (CO):

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

- **CO1:** Explain the physical and logical design of IoT, enabling technologies, deployment levels, and applications across different domains.
- **CO2:** Differentiate IoT and M2M communication and explain the roles of SDN, NFV, SNMP, and NETCONF-YANG in IoT system management.
- **CO3:** Apply IoT design methodology and Python programming concepts to develop the logical components of an IoT solution.
- **CO4:** Interface IoT devices with Raspberry Pi and develop Python programs for sensing, data acquisition, and device control.
- **CO5:** Design and implement an IoT application for home automation, smart agriculture, smart cities, or smart healthcare.

UNIT – I

Introduction to Internet of Things, Introduction, physical design, logical design, IoT enabled technologies, IoT levels & deployment templates. Domain specific IoTs, Introduction, home automation, cities, environment, energy, retail, logistics, agriculture, Industry and health & lifestyle.

UNIT – II

IoT and M2M, Introduction, M2M, difference between IoT and M2M, SDN and NFV for IoT, IoT system management with NETCONF-YANG, need for IoT systems management, SNMP, network operator requirements, ETCONF, YANG.

UNIT – III

IoT platforms design methodology, Introduction, IoT design methodology, case study, motivation for using Python. IoT Systems – Logical design using python, introduction, python data types and structures, control flow, functions, modules, packages, file handling, date/time operations, classes and packages.

UNIT – IV

IoT physical Devices and endpoints, IoT devices, Raspberry Pi, Raspberry Pi interfaces, programming Raspberry Pi with Python. Case Studies Illustrating IoT Design, home automation, cities, environment and agriculture & productivity applications.

UNIT V

APPLICATIONS DEVELOPMENT: Complete Design of Embedded Systems – Development of IoT Applications – Home Automation – Smart Agriculture – Smart Cities – Smart Healthcare.

Prescribe Textbook

- Internet of Things – A Hands-On Approach, Arsdeep Bahga & Vijay Madisetti, Universities Press

Reference Books

- The Internet of Things Enabling Technologies, Platforms, and Use Cases, Pethuru Raj and Anupama C. Raman, CRC Press.
- IoT Fundamentals Networking Technologies, Protocols and Use Cases for the Internet of Things , David, Hanes & Salgueiro Gonzalo, Pearson Education.

Blockchain Technology					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26305D	**PE – IV**	3	1	0	4	30	70	100

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.
LO5	To make them familiar with payments with Bitcoin network

Course Outcomes (CO):

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

CO1	Know the basics of Blockchain Terminology.
CO2	Understand the concept of Blockchain, Bitcoin
CO3	Understand de-centralization
CO4	Gain knowledge about the domain of blockchain in real time.
CO5	How to do payments with Bitcoin network.

UNIT – I

Blockchain, the growth of blockchain technology, distributed systems, the history of blockchain and Bitcoin, types of blockchains

UNIT –II

Decentralization, methods of decentralization, routes of decentralization, blockchain and full ecosystem decentralization, smart contracts, Decentralized organizations and platforms for decentralization.

UNIT – III

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 – IV

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

UNIT – V

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

Prescribe Book

- 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.

Advanced Java Programming Lab					M.Sc. 2nd year 3rd semester			
Code	Category	Hours / Week			Credits	MARKS		
		L	T	P	C	Internal	End Exam	Total
MS26306	PC – LAB	0	0	6	3	25	50	75

1. Develop and demonstrate a HTML document that illustrates the use external style sheet, ordered list, table, borders, padding, color, and the tag.
2. Write HTML code to provide intra document linking.
3. Create a form with the following specifications

a) Our form uses frames, one to hold the links bar at the top of the browser window.

b) Other is a larger frame that provides the main view.

c) The links bar should contain 5 links, which when clicked, should display the appropriate HTML file in the larger frame.

4. to create a webpage with the following using html

a. to embed an image in web page

b. to fix the hot spots

c. show all the related information when a hot spot is clicked in the map

5. Develop a HTML Form, which accepts any Mathematical expression. Write JavaScript code to Evaluates the expression and Displays the result.
6. Create a HTML form that has number of Textboxes. When the form runs in the Browser fill the textboxes with data. Write JavaScript code that verifies that all textboxes has been filled. If a textboxes has been left empty, popup an alert indicating which textbox has been left empty.
7. Write a JavaScript code to find the sum of N natural Numbers. (Use user-defined function)
8. Write a JavaScript code to find factorial of N. (Use recursive function)
9. Write a JavaScript code block using arrays and generate the current date in words, this should include the day, month and year.
10. Create a web page using two image files, which switch between one another as the mouse pointer moves over the images. Use the onMouseOver and onMouseOut event handlers.
11. Design an XML document to store information about a student in an engineering college affiliated to ANU. The information must include college id, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and useit to display the document.
12. Create an XML document, which contains 10 users information. Implement a program, which takes User Id as an input and returns the user details by taking the user information from the XML document
13. write a program for implementing student information using XML
14. write a java program to illustrate java to database connectivity using JDBC
15. Write a program to print the Fibonacci numbers using RMI.
16. Write a program using RMI to access the database using the primary key value and return the data to the client.

17. Write a html program for invoking servlet from applet
18. write a java servlet program to conduct online examination and to display student mark list available in a database
19. Create a java program to create an airline reservation service and a travel agent and the travel agent is searching for an airline using web services and database.
20. Write a JSP program to calculate income tax, login and data capture.

Note: The above experiments are for indicative purposes only. However, the concerned faculty member can add a few more experiments in addition to the existing. In such cases the concerned faculty member should get the syllabus approved by the BoS.

Machine Learning Lab					M.Sc. 2nd year 3rd semester			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26307	**PC – LAB**	0	0	6	3	25	50	75

1.	The probability that it is Friday and that a student is absent is 3%. Since there are 5 school days in a week, the probability that it is Friday is 20%. What is the probability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result.(Ans: 15%)
2.	Extract the data from database using python
3.	Implement k-nearest neighbors classification using python
4.	Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)
5.	The following training examples map descriptions of individuals onto high, medium and low creditworthiness. Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of 'golf' and the conditional probability of 'single' given 'medRisk' in the dataset
6.	Implement linear regression using python
7.	Implement naive baye's theorem to classify the English text
8.	Implement an algorithm to demonstrate the significance of genetic algorithm
9.	Implement the finite words classification system using Back-propagation algorithm
10	Additional Experiments: Find-S and Candidate Elimination Algorithms

Note: The above experiments are for indicative purposes only. However, the concerned faculty member can add a few more experiments in addition to the existing. In such cases the concerned faculty member should get the syllabus approved by the BoS.

R Programming Lab					M.Sc. 2nd year 1st semester			
Code	**Category**	**Hours / Week**			**Credits**	**MARKS**		
		L	**T**	**P**	**C**	**Internal**	**End Exam**	**Total**
MS26308	**PC – LAB**	-	-	4	2	50	-	50

1. Download and install R-Programming environment and install basic packages using install.

Packages () command in R.

2. Learn all the basics of R-Programming (Data types, Variables, Operators etc.)
3. Implement R-Loops with different examples.
4. Learn the basics of functions in R and implement with examples.
5. Implement data frames in R. Write a program to join columns and rows in a data frame using c bind () and r bind () in R.
6. Implement different String Manipulation functions in R.
7. Implement different data structures in R(Vectors ,Lists ,Data Frames)
8. Write a program to read acsv file and analyze the data in the file in R
9. Create pie charts and bar charts using R.
10. Create a data set and do statistical analysis on the data using R.
11. Write R program to find Correlation and Covariance
12. Write R program for Regression Modeling
13. Write R program to build classification model using KNN algorithm
14. Write R program to build clustering model using K-mean algorithm

Note: The above experiments are for indicative purposes only. However, the concerned faculty member can add a few more experiments in addition to the existing. In such cases the concerned faculty member should get the syllabus approved by the BoS.

SWAYAM Plus Courses					M.Sc. 2nd year 3rd semester			
Code	Category	Hours / Week			Credits	MARKS		
		L	T	P	C	Internal	End Exam	Total
MS26308	PC – LAB	-	-	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):

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

M.Sc. II Year – IV Semester

S. No.	Course Code	Category	Course	Max Marks		Hours Per Week			Credits
				I	E	L	T	P	
1	MS26401	PROJ	Major Project	100	200	0	0	36	12
			TOTAL	**100**	**200**	**0**	**0**	**36**	**12**

Project Work					M.Sc. 2nd year 4th semester			
Code	Category	Hours / Week			Credits	MARKS		
		L	T	P	C	Internal	End Exam	Total
MS26401	PROJ	0	0	36	12	100	200	300

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):

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 should not substitute for the required project work, reviews, and assessment.