CHOICE BASED CREDIT SYSTEM

EVALUATION SCHEME

AND

COURSE OF STUDY

IN

M.TECH

COMPUTER SCIENCE AND ENGINEERING

(I YEAR & II YEAR)

SCHEME OF EXAMINATION & SYLLABUS

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY
GURUKULA KANGRI (DEEMED TO BE UNIVERSITY), HARIDWAR

Faculty of Engineering & Technology

In the year 2000 Faculty of Engineering & Technology was established with an aim of imparting technical education in the spiritual surroundings of the Gurukula System. Keeping in mind the importance of technocrats with strong moral character, superior knowledge, and devotion to the nation. FET was established with a motto of Building Technocrats with ethics. FET is known in India and abroad for students with virtuous moral character and Technical abilities. FET is one of the richest faculty of Gurukula Kangri (Deemed to be University), with a huge number of books in the library, well-equipped electronics electrical and mechanical laboratories, latest software, and computers in computer labs. Football field, Tennis court, Volleyball court, Basketball arena, and open gym for the students with athletic interests.

Vision of F.E.T.

To provide affordable & quality education to engineering aspirants and nurture them to be highly skilled & innovative technocrats with ethics and nation building spirit.

Mission of F.E.T.

M1: (ETHICS & VALUES)
To educate and nurture engineering aspirants with values, updated engineering curriculum & latest technology to make them globally trusted and accepted.

M2: (RESEARCH)
Provide conducive environment for teaching, learning & research that can lead to patents, publications and make country proud.

M3: (AFFORDABILITY)
Provide cost effective education so that every section of society can be benefitted.

M4: (SKILLED)
Design industry oriented curriculum that can make engineering graduates ready to work for Indian Industries as well as MNCs.

Department of Computer Science & Engineering

The Department of Computer Science and Engineering (CSE) provides in-depth technical knowledge and opportunities for innovation and research with the latest computer facilities.

Vision And Mission

Vision of the department

To be a frontier in the field of Computer Science by imparting the knowledge in legible, lucid and perspicuous way and preparing the human resource of high moral and ethical values that can cater to contemporary societal needs.

Mission of the department

- **[M1]: (Contemporary excellence)**
  Provide a sound technical foundation in Computer Engineering through the comprehensive curriculum with a rich skill set and practical experience.
- **[M2]: (Holistic Learning)**
  To enable students to become valuable and creative contributors to society. To continue their education in different facets of technology to grow them professionally along with the spirit of moral values.
- **[M3]: (Social Responsibility & Sustainable Development)**
  To contribute to National Development by meeting the needs of society and industry, empowering weaker and underprivileged sections, and building the economy through research and frugal innovation, anchored in the principle of achieving more with less.
- **[M4]: (Ethics & Values)**
  To uphold the highest ethical standards, inculcate values; create willingness and capacity to work with one’s hands, and a spirit of devotion to serve humanity.

Master of Technology (Computer Science & Engineering)_ M. Tech (CSE)_

M. Tech stands for Master of Technology. It's a postgraduate program in engineering and technology fields pursued after a Bachelor's degree in a related field, typically lasting two years.

1. Introduction

The Department of Computer Science and Engineering at FET, Gurukul Kangri, Haridwar starts a new M.Tech program in Computer Science and Engineering (CSE) with an intake of 18 students from session 2024-25. This program aims to equip graduates with advanced knowledge and skills in critical areas of CSE to address the ever-growing demand for qualified professionals in the industry.

2. Program Structure

The M. Tech Program will be a four-semesters (Two-years) program. Students will complete course work, laboratory sessions, and a final project/thesis.

The M. Tech in CSE program will be a two-years, full-time program consisting of:

- **Core Courses:** These courses provide a strong foundation in advanced computer science concepts.
- **Elective Courses:** Students can choose electives to specialize in specific areas of interest.
- **MOOC Courses:** Students will attend online lectures/seminars on advanced topics in computer science.
- **Master's Thesis:** Students will undertake a research project under the supervision of a faculty member.

3. Admission Process/Eligibility Criteria

The program will consider applications from both GATE (Graduate Aptitude Test in Engineering) qualified and non-GATE students, though preference will be given to candidates having valid GATE score. Eligible candidates will be shortlisted based on their GATE score/Last Eligible Degree Percentage. The university may consider factors such as overall CGPA or percentage in B.Tech./UG program. Candidates with valid GATE score are eligible for AICTE scholarship as per AICTE norms. The admission criteria may vary as per AICTE/University norms.

Students must possess a Bachelor's degree in Engineering (BE/B. Tech) in CSE/IT/SE/ECE/EEE or allied branches from a recognized university. An MCA (Master of Computer Applications)/M. Sc (CS/IT), (with Mathematics at B. Sc/BCA level) are also eligible.This may vary as per AICTE/University norms. The university will set a minimum qualifying GATE score or UG exam percentile based on the program's competitiveness.

4. Program Outcomes of CSE (M. Tech.) program:

The main outcomes of the CSE (M. Tech.) program are given here. At the end of the program a student is expected to have:

1. An understanding of the theoretical foundations and the limits of computing.
2. An ability to adapt existing models, techniques, algorithms, data structures, etc. for efficiently solving problems.
3. An ability to design, develop and evaluate new computer based systems for novel applications which meet the desired needs of industry and society.
4. Understanding and ability to use advanced computing techniques and tools.
5. An ability to undertake original research at the cutting edge of computer science & its related areas.
6. An ability to function effectively individually or as a part of a team to accomplish a stated goal.
7. An understanding of professional and ethical responsibility.
8. An ability to communicate effectively with a wide range of audience.
9. An ability to learn independently and engage in life¬long learning.
10. An understanding of the impact of IT related solutions in an economic, social and environment context.

5. Course Structure

Year I - Semester 1

DSC/SEC/DSE/AEC	SUBJECT	PERIODS			EVALUATION SCHEME				Subject Total	Credits
					SESSIONAL EVALUATION			EXAM		
		L	T	P	CT	TA	Total	ESE		
THEORY										
MCE-C101	Program Core I- Mathematical foundations of Computer Science	3	0	0	20	10	30	70	100	3
MCE-C102	Program Core II- Advanced Data Structures	3	0	0	20	10	30	70	100	3
MCE-E10X	Program Elective I (From List I,II and III)	3	0	0	20	10	30	70	100	3
MCE-S104	MOOC	3	0	0	20	10	30	70	100	3
MCE-A104	Research Methodology and IPR	2	0	0	20	10	30	70	100	2
PRACTICAL										
MCE-C155	Laboratory 1 (Advanced Data Structures Lab)	0	0	4	10	5	15	35	50	2
MCE-E15X	Laboratory 2 (Based on Electives)	0	0	4	10	5	15	35	50	2
TOTAL		14	0	8	120	60	180	420	600	18

Year I - Semester II

DSC/SEC/DSE/AEC	SUBJECT	PERIODS			EVALUATION SCHEME				Subject Total	Credits
					SESSIONAL EVALUATION			EXAM		
		L	T	P	CT	TA	Total	ESE		
THEORY										
MCE-C201	Program Core III – Advance Algorithms	3	0	0	20	10	30	70	100	3
MCE-C202	Program Core IV – Soft Computing	3	0	0	20	10	30	70	100	3
MCE-E20X	Program Elective II (From List I,II and III)	3	0	0	20	10	30	70	100	3
MCE-S204	MOOC	3	0	0	20	10	30	70	100	3
PRACTICAL										
MCE-C255	Laboratory 3 (Based on cores)	0	0	4	10	5	15	35	50	2
MCE-C256	Laboratory 4 (Based on Electives)	0	0	4	10	5	15	35	50	2
MCE-C260	Minor Project	2	0	0	10	5	15	35	50	2
TOTAL		14	0	8	110	55	165	385	550	18

Year II - Semester III

MCE	SUBJECT	PERIODS			EVALUATION SCHEME		Subject Total	Credits
					SESSIONAL EVALUATION	EXAM ESE		
		L	T	P				
THEORY								
MCE-P390	Minor Project /INTERNSHIP/ Industry Project	0	0	6	300	-	300	8
MCE-S391	Seminar	0	0	4	100	-	100	4
	Total	0	0	10	400	-	400	12

Year II - Semester IV

MCE	SUBJECT	PERIODS			EVALUATION SCHEME		Subject Total	Credit
					SESSIONAL EVALUATION	EXAM ESE		
		L	T	P				
THEORY								
MCE-P490	Dissertation with Research paper / INTERNSHIP/ Industry Project	0	0	16	100	400	500	16
	Total	0	0	16	100	400	500	16

Credit				
Semester	**1**	**2**	**3**	**4**
Semester-wise Total Credits	18	18	12	16
Total Credits	**64**			

Elective 1

S. No.	CODE	Course Name
1	MCE-E103	Block chain and its applications
2	MCE-E104	Computer Vision and Image processing
3	MCE-E105	Distributed Computing
4	MCE-E106	Advance Data Science
5	MCE-E107	Big Data Analytics
6	MCE-E108	Machine Learning
7	MCE-E156	Block chain and its applications lab

Elective 2

S. No	CODE	Course Name
1	MCE-E201	Information and Network Security
2	MCE-E202	Ethical Hacking
3	MCE-E203	Parallel computing
4	MCE-E204	Internet of Things
5	MCE-E205	Advance Artificial Intelligence
6	MCE-E206	Genetic Algorithms

Elective 3

S. No	CODE	Course Name
1	MCE-E109 / MCE-E208	Software Testing and Auditing
2	MCE-E110 / MCE-E209	Mobile Application Development
3	MCE-E111 / MCE-E210	Scientific Computing with Python
4	MCE-E112 / MCE-E211	Quantum Computing
5	MCE-E113 / MCE-E212	Robotics
6	MCE-E114 / MCE-E213	Cloud Computing

Note:

List of MOOC courses shall be decided by the departmental committee in each semester depending upon the list from SWAYAM/NPTEL and other recognized online platforms. Students have to study from Online Platform doubt sessions shall be held by Internal teachers and exams shall be taken by university. If a student wishes he can give an exam of Online Platform for certification. SWAYAM courses to run every year from July onwards (Odd Semester) are declared in the month of May and for courses to run every year from January onwards (Even Semester) are declared in the month of December on website https://swayam.gov.in/.

Syllabus

Course Code	MCE-C101
Course Name	Mathematical Foundation of Computer Science
Credits	3
Pre-Requisites	Discrete Structures

Total Number of Lectures: 35

COURSE OBJECTIVE
• To understand the mathematical fundamentals that is prerequisites for a variety of courses like Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems, Bioinformatics, Machine learning.
• To develop the understanding of the mathematical and logical basis to many modern. techniques in information technology like machine learning, programming language design, and concurrency.
• To study various sampling and classification problems.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 Probability mass, density, and cumulative distribution functions, Parametric families of distributions, Expected value, variance, conditional expectation, Applications of the univariate and multivariate Central Limit Theorem, Probabilistic inequalities, Markov chains	5
Unit 2 Random samples, sampling distributions of estimators**,** Methods of Moments and Maximum Likelihood**,**	5
Unit 3 Statistical inference**,** Introduction to multivariate statistical models: regression and classification problems, principal components analysis**,** The problem of overfitting model assessment.	7
Unit 4 Graph Theory: Isomorphism, Planar graphs, graph colouring, hamilton circuits and euler cycles. Permutations and Combinations with and without repetition. Specialized techniques to solve combinatorial enumeration problems	7
Unit 5 Computer science and engineering applications, Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems, Bioinformatics, Machine learning.	7
Unit 6 Recent Trends in various distribution functions in the mathematical field of computer science for varying fields like bioinformatics, soft computing, and computer vision.	4

COURSE OUTCOMES

After completion of course, students would be able to:

- To understand the basic notions of discrete and continuous probability.
- To understand the methods of statistical inference, and the role that sampling distributions play in those methods.

- To be able to perform correct and meaningful statistical analyses of simple to moderate complexity.

References

1. John Vince, Foundation Mathematics for Computer Science, Springer.
2. K. Trivedi.Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Wiley.
3. M. Mitzenmacher and E. Upfal.Probability and Computing: Randomized Algorithms and Probabilistic Analysis.
4. Alan Tucker, Applied Combinatorics, Wiley

Course Code	MCE-C102
Course Name	Advanced Data Structures
Credits	3
Pre-Requisites	Data Structures

Total Number of Lectures : 35

COURSE OBJECTIVE
• The student should be able to choose appropriate data structures, understand the ADT/libraries, and use it to design algorithms for a specific problem. • Students should be able to understand the necessary mathematical abstraction to solve problems. • To familiarize students with advanced paradigms and data structure used to solve algorithmic problems. • Students should be able to come up with analysis of efficiency and proofs of correctness.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Dictionaries:** Definition, Dictionary Abstract Data Type, Implementation of Dictionaries. **Hashing:** Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.	4
Unit 2 **Skip Lists:** Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists	6
Unit 3 **Trees:** Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees	5
Unit 4 **Text Processing:** Sting Operations, Brute-Force Pattern Matching, The Boyer- Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.	7
Unit 5 **Computational Geometry:** One Dimensional Range Searching, Two Dimensional Range Searching, Constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quadtrees, k-D Trees.	8
Unit 6 Recent Trends in Hashing, Trees, and various computational geometry methods for efficiently solving the new evolving problem	5

COURSE OUTCOMES
• After completion of course, students would be able to: • Understand the implementation of symbol tables using hashing techniques. • Develop and analyze algorithms for red-black trees, B-trees and Splay trees. • Develop algorithms for text processing applications. Identify suitable data structures and develop algorithms for computational geometry problems.

References:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004.
2. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.

Course Code	MCE-E103
Course Name	Blockchain and its Applications
Credits	3
Pre-Requisites	Basic networking and distributed system concepts

Total Number of Lectures : 35

COURSE OBJECTIVE

- Provide a comprehensive understanding of blockchain technology, its history, evolution, and core principles like distributed ledgers and consensus mechanisms.
- Explore different blockchain architectures (public, private, permissioned) and their suitability for various applications
- Expose students to real-world applications of blockchain technology in various sectors, with a focus on the finance industry (cryptocurrencies, DeFi, etc.).
- Introduce students to the concept of smart contracts, their functionalities, and potential use cases within blockchain ecosystems.
- Broaden the understanding of blockchain applications beyond finance, exploring its potential in supply chain management, healthcare, identity management, and governance

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 **Introduction to Blockchain Technology:** What is Blockchain? History and evolution of blockchain technology. Decentralized Ledgers: Understanding the core concept of distributed trust. Cryptography for Blockchain: Introduction to cryptographic primitives (hashing, digital signatures) used in blockchain.	8
Unit 2 **Consensus Mechanisms:** Proof of Work (PoW), Proof of Stake (PoS), and other consensus algorithms for blockchain networks.	7
Unit 3 **Blockchain Architectures and Smart Contracts:** Public vs. Private vs. Permissioned Blockchains: Different blockchain architecture types and their applications. Smart Contracts: Introduction to smart contracts, their functionalities, and use cases. Programming Languages for Blockchain: Exploring languages like Solidity for writing smart contracts. Decentralized Applications (dApps): Building and deploying applications on top of blockchain platforms.	7
Unit 4 **Applications of Blockchain in Finance:** Cryptocurrencies and Tokenization: Understanding Bitcoin, Ethereum, and other cryptocurrencies. Decentralized Finance (DeFi): Exploring DeFi applications like lending, borrowing, and trading on blockchain. Security Token Offerings (STOs): How blockchain can revolutionize traditional financial instruments. Trade Finance and Cross-Border Payments: Streamlining trade finance processes with blockchain.	5
Unit 5 **Applications of Blockchain Beyond Finance :** Supply Chain Management: Enabling transparency and traceability in supply chains. Healthcare: Utilizing blockchain for secure patient data management and drug tracking. Identity Management: Building secure and verifiable digital identities on blockchain. Governance and Voting Systems: Enhancing transparency and security in elections and voting processes.	4
Unit 6 **Future of Blockchain and Emerging Trends:** Scalability Challenges and Solutions: Exploring scalability issues and potential solutions for blockchain technology. Blockchain Regulations and Legal Considerations: The evolving regulatory landscape for blockchain applications. The Social	4

Impact of Blockchain: Exploring the potential societal implications of blockchain adoption. Emerging Trends in Blockchain: Discussing recent advancements and future directions in blockchain technology.	

COURSE OUTCOMES

After completion of course, students would be able to:

- Students will be able to explain the concept of blockchain technology, its historical development, and the core principles of distributed ledgers and decentralization.
- Students will be able to define smart contracts, explain their functionalities, and identify potential applications in various scenarios.
- Students will be able to analyze the role of blockchain technology in revolutionizing the financial sector, including cryptocurrencies, DeFi, and security token offerings
- Students will be able to critically evaluate the scalability challenges faced by blockchain technology and explore potential solutions for addressing them.
- Students will be able to identify and analyze emerging trends in blockchain technology and its future directions.

References:

1. Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World by Don Tapscott and Alex Tapscott

Course Code	MCE-E104
Course Name	Computer Vision and Image Processing
Credits	3
Pre-Requisites	Computer graphics

Total Number of Lectures : 35

COURSE OBJECTIVE
• Provide a comprehensive understanding of fundamental digital image representation, processing techniques, and feature extraction methods. • Introduce students to image restoration concepts, morphological image processing, and feature matching techniques. • Equip students with the knowledge of camera models, object detection strategies, and feature-based object recognition approaches. • Foster an understanding of how machine learning and deep learning can be applied for computer vision tasks.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 **Fundamentals of Digital Images :** Digital Image Representation: Pixel formats, color spaces (RGB, grayscale), image resolution. Image Enhancement Techniques: Contrast stretching, histogram equalization, noise reduction filters. Image Filtering: Spatial domain filtering (averaging, median, Gaussian), frequency domain filtering. Image Segmentation: Thresholding, region-based segmentation, edge detection (Canny edge detector).	7
Unit 2 **Image Restoration and Feature Extraction** : Image Restoration Techniques: Noise modeling, filtering for noise reduction, deblurring techniques. Morphological Image Processing: Erosion, dilation, opening, closing for shape analysis. Feature Extraction Techniques: Edge features, shape descriptors (moments, Hu invariants), texture analysis. Feature Matching and Correspondence: Techniques for establishing relationships between features in images.	5
Unit 3 **Introduction to Computer Vision :** Camera Models and Image Formation: Pinhole camera model, perspective projection, geometric distortions. Object Detection and Recognition: Sliding window approach, template matching, Viola-Jones framework. Feature-Based Object Recognition: SIFT, SURF features, object recognition pipelines. Machine Learning for Computer Vision: Introduction to using machine learning for image classification and object recognition.	6
Unit 4 Convolutional Neural Networks (CNNs): Architecture, convolutional layers, pooling layers, activation functions. Training CNNs for Image Classification: Techniques for training CNNs on image datasets (e.g., ImageNet). CNN Architectures for Object Detection: YOLO, R-CNN, Single Shot MultiBox Detector (SSD). Semantic Segmentation with Deep Learning: Fully convolutional networks (FCNs) for pixel-wise labeling.	6
Unit 5 Computer Vision Applications: Medical imaging analysis, facial recognition, autonomous vehicles. Performance Evaluation Metrics: Precision, recall, F1-score, Intersection over Union (IoU). Emerging Trends in Computer Vision: Generative Adversarial Networks (GANs) for image synthesis, visual question answering.	7
Unit 6 Ethical Considerations in Computer Vision: Bias in algorithms, privacy concerns, responsible development practices.	4

COURSE OUTCOMES
After completion of course, students would be able to: • Students will be able to analyze digital image characteristics, apply enhancement techniques, and perform basic image filtering operations in both spatial and frequency domains. • Students will be able to implement image restoration techniques for noise reduction and deblurring, and utilize morphological operations for shape analysis. • Students will be able to design and train basic machine learning models for image classification tasks and understand their application in computer vision. • Students will be able to explain the architecture and functionality of Convolutional Neural Networks (CNNs), train CNNs for image classification, and analyze different CNN architectures for object detection and semantic segmentation.

References:

1. Computer Vision: Algorithms and Applications by Richard Szeliski.
2. Deep Learning, by Goodfellow, Bengio, and Courville.
3. Dictionary of Computer Vision and Image Processing, by Fisher et al.
4. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Fourth Edition or later)

Course Code	MCE-E105
Course Name	Distributed Computing
Credits	3
Pre-Requisites	Operating Systems

Total Number of Lectures:35

COURSE OBJECTIVE
• Provide a comprehensive understanding of distributed system architectures, communication models, and fundamental challenges like consistency and concurrency control. • Introduce students to distributed consensus protocols, replication techniques, and design considerations for distributed file systems. • Expose students to various distributed programming paradigms like RPC, message-oriented middleware, and distributed object systems, along with the principles of cloud-based distributed computing. • Foster the ability to design, implement, and secure scalable distributed applications, considering distributed databases, transactions, and security best practices.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to Distributed Systems:** What is Distributed Computing? Characteristics, challenges, and benefits of distributed systems. Architectures for Distributed Systems: Client-server, peer-to-peer, distributed object systems. Communication Models: Message passing vs. shared memory, synchronous vs. asynchronous communication. Distributed Naming and Consistency: Naming services, replica management, consistency models.	7
Unit 2 **Concurrency Control and Synchronization:** Concurrency Control Problems: The race condition problem, critical sections, mutual exclusion. Software Synchronization Techniques: Semaphores, monitors, lock-based synchronization. Distributed Mutual Exclusion Algorithms: Lamport's Bakery algorithm, token-based algorithms. Distributed Deadlock Detection and Recovery: Detecting deadlocks, deadlock avoidance, recovery strategies.	6
Unit 3 **Fault Tolerance and Replication:** Fault Tolerance in Distributed Systems: Types of failures, Byzantine failures, fault tolerance mechanisms. Distributed Consensus Protocols: Reaching agreement in a distributed system (Paxos, Raft). Replication Techniques: Active and passive replication, state management in replicated systems. Distributed File Systems: Challenges of distributed file systems, consistency protocols (Coda, HDFS).	7
Unit 4 **Distributed Programming Paradigms:** Remote Procedure Calls (RPC): Principles, design considerations, RPC frameworks. Message-Oriented Middleware (MOM): Introduction to message queues (e.g., RabbitMQ, Apache Kafka). Distributed Objects: Distributed object systems (CORBA, Java RMI), remote method invocation. Cloud-Based Distributed Computing: Leveraging cloud platforms for distributed applications.	7
Unit 5 Building Scalable Distributed Applications: Design patterns, communication strategies, performance considerations. Distributed Database Systems: Challenges and design principles for distributed databases.	4
Unit 6 Distributed Transactions: Atomicity, consistency, isolation, durability (ACID) properties in distributed transactions. Security in Distributed Systems: Authentication, authorization, access control in distributed environments.	4

COURSE OUTCOMES
After completion of course, students would be able to: • Students will be able to differentiate between various distributed system architectures, analyze communication models, and explain the challenges associated with distributed computing. • Students will be able to implement software synchronization techniques, design and analyze distributed mutual exclusion algorithms, and propose strategies for deadlock detection and recovery. • Students will be able to design scalable distributed applications, analyze challenges in distributed databases, and implement solutions for ensuring transaction properties (ACID) in distributed settings. • Students will be able to develop distributed applications using remote procedure calls (RPC), utilize message-oriented middleware, and leverage distributed object systems for building distributed functionalities.

References:

1. Distributed Systems: Concepts and Design by George Coulouris, Jean Dollimore, and Tim Kindberg (Fifth Edition or later).
2. Designing Data-Intensive Applications: The Big Data Stack by Martin Kleppmann.
3. Understanding Distributed Systems by Roberto Vitillo.

Course Code	MCE-E106
Course Name	Advance Data Science
Credits	3
Pre-Requisites	Basics of Machine learning and deep learning

Total Number of Lectures : 35

COURSE OBJECTIVE
● Equip students with advanced Deep Learning architectures like LSTMs, Transformers, and GANs for sequential data, NLP tasks, and generative modeling. ● Introduce students to scalable machine learning techniques for handling big data and explore model selection, ensemble methods, and distributed training. ● Foster advanced feature engineering practices, including feature selection and importance measures, tailored for different data types. ● Provide a platform for independent research and exploration of advanced topics in Deep Learning and Machine Learning.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Advanced Deep Learning Architectures:** Recurrent Neural Networks (RNNs) for Sequential Data: Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), sequence-to-sequence learning. Attention Mechanisms: Understanding and applying attention mechanisms in neural networks. Transformers and Deep Learning for NLP: Transformer architecture, applications in language modeling, machine translation (e.g., BERT, GPT-3). Generative Adversarial Networks (GANs): Advanced GAN architectures, applications in image generation, and style transfer.	7
Unit 2 **Scalable Machine Learning:** Big Data Processing Frameworks: Spark, Hadoop, and distributed computing for large-scale data processing. Model Selection and Ensemble Methods: Cross-validation techniques, hyperparameter tuning, ensemble methods (e.g., Random Forests, Gradient Boosting) for improved performance. Distributed Machine Learning Systems: Scaling machine learning models using frameworks like TensorFlow Distributed or PyTorch Lightning. Model Compression and Quantization: Techniques for reducing model size and resource footprint for deployment.	6
Unit 3 **Unsupervised Learning and Dimensionality Reduction:** Advanced Clustering Techniques: K-Means variations, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Hierarchical clustering. Dimensionality Reduction Techniques: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), dimensionality reduction for visualization and feature engineering. Autoencoders and Variational Autoencoders (VAEs): Learning latent representations with autoencoders, VAEs for unsupervised generative modeling. Anomaly Detection Techniques: Statistical anomaly detection, one-class Support Vector Machines (SVMs) for identifying outliers.	7
Unit 4 **Advanced Feature Engineering and Feature Selection:** Feature Engineering Techniques: Feature scaling, normalization, feature hashing for categorical data, creating informative features from raw data. Feature Selection and Importance Measures: Filter methods, wrapper methods, embedded methods for selecting relevant features. Feature Engineering for Specific Applications: Techniques tailored for text data, image data, time series data. Feature Stores and Feature Management: Introduction to feature stores for managing large feature sets in production environments.	8

Unit 5 **Advanced Topics and Research:** Explainable AI (XAI): Techniques for understanding and interpreting predictions from complex models. Responsible Data Science: Bias in data and models, fairness considerations, data privacy challenges.	4
Unit 6 Advanced Optimization Techniques: AdamW, RMSprop optimizers, gradient clipping, learning rate scheduling for better model training.	3

COURSE OUTCOMES

After completion of course, students would be able to:

- Students will be able to design and implement advanced Deep Learning architectures (LSTMs, Transformers, GANs) for various tasks like sequence modeling, language processing, and image generation.
- Students will be able to leverage big data processing frameworks (Spark, Hadoop) and distributed machine learning systems for training models on large-scale datasets.
- Students will be able to critically evaluate Explainable AI (XAI) techniques, discuss responsible data science practices, and implement advanced optimization algorithms for efficient model training.
- Students will be able to conduct independent research on an advanced AI topic, present their findings, and critically evaluate research papers in the field.

References:

1. Taming the Big Data Tidal Wave: Finding Opportunities in the Flood of Information by Bill Franks

Course Code	MCE-E107
Course Name	Big Data Analytics
Credits	3
Pre-Requisites	SQL for relational databases

Total Number of Lectures : 35

COURSE OBJECTIVE
● Equip students with a comprehensive understanding of Big Data concepts, challenges, and opportunities associated with the 3 Vs (Volume, Velocity, Variety). ● Introduce students to the fundamental principles of Big Data analytics, data lifecycle management, data warehousing vs. data lakes, and data governance practices. ● Foster hands-on skills in working with distributed file systems (HDFS), distributed processing frameworks (MapReduce, Spark), and NoSQL databases for handling Big Data. ● Expose students to Big Data analytics tools and techniques, including data visualization, scalable machine learning algorithms, stream processing with Kafka, and cloud-based Big Data analytics platforms.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to Big Data:**What is Big Data? The 3 Vs of Big Data (Volume, Velocity, Variety), challenges and opportunities. Big Data Analytics Concepts: Data lifecycle, data warehousing vs. data lakes, data governance principles. Big Data Applications: Big Data analytics across various industries (e.g., finance, healthcare, retail). Big Data Ecosystem Introduction: Overview of key technologies and frameworks for Big Data (Hadoop, Spark, cloud platforms).	8
Unit 2 **Big Data Storage and Processing:** Distributed File Systems (DFS): Introduction to HDFS (Hadoop Distributed File System), data management in HDFS. Distributed Processing Frameworks: MapReduce paradigm, Hadoop YARN (Yet Another Resource Negotiator), resource management for Big Data jobs. Apache Spark: Spark architecture, Resilient Distributed Datasets (RDDs), in-memory processing capabilities. NoSQL Databases for Big Data: Introduction to NoSQL databases (e.g., MongoDB, Cassandra), advantages for unstructured and semi-structured data.	6
Unit 3 **Big Data Processing with Hadoop:** Hadoop Ecosystem Tools: HDFS commands, data ingestion with Flume, data querying with Hive (Hadoop HiveQL). MapReduce Programming: Writing and deploying MapReduce jobs for data processing tasks (e.g., word count). Apache Pig: High-level data flow language for Pig Latin, simplifying MapReduce development. Big Data Security: Security considerations and best practices for Big Data storage and processing.	7
Unit 4 **Big Data Analytics Tools and Techniques:** Big Data Visualization: Techniques for visualizing Big Data (e.g., with Apache Zeppelin, Tableau) for effective communication of insights. Machine Learning for Big Data: Scalable machine learning algorithms for Big Data analysis (e.g., Spark MLlib). Stream Processing with Apache Kafka: Real-time data processing with Kafka, applications in streaming analytics. Big Data Analytics on Cloud Platforms: Leveraging cloud platforms (e.g., AWS, Azure, GCP) for Big Data storage, processing, and analytics.	6
Unit 5: **Big Data Project & security** Big Data Project Development: Students work in teams to design and implement a Big Data analytics solution for a chosen problem domain. Big Data Governance and Regulations: Data privacy regulations (e.g., GDPR), data quality management in Big Data environments.	4

Emerging Trends in Big Data: Big Data and Artificial Intelligence (AI), Internet of Things (IoT) and Big Data, ethical considerations in Big Data. Big Data Careers and Industry Landscape: Career opportunities in Big Data analytics, industry trends and future directions.	
Unit 6 **Big Data Future Trends:** Emerging Trends in Big Data: Big Data and Artificial Intelligence (AI), Internet of Things (IoT) and Big Data, ethical considerations in Big Data. Big Data Careers and Industry Landscape: Career opportunities in Big Data analytics, industry trends and future directions.	4

COURSE OUTCOMES

After completion of course, students would be able to:

- Students will be able to differentiate between data warehousing and data lakes, understand data governance principles, and identify key technologies within the Big Data ecosystem (Hadoop, Spark, cloud platforms).
- Students will be able to design and implement a Big Data analytics solution for a chosen problem domain through a collaborative project.
- Students will be able to understand the MapReduce paradigm, write and deploy MapReduce jobs for data processing tasks, and leverage Apache Pig for simplifying MapReduce development.
- Students will be able to explain data privacy regulations (e.g., GDPR), discuss data quality management in Big Data environments, and analyze emerging trends like Big Data and AI, IoT and Big Data, and ethical considerations.

References:

1. Big Data: A Revolution That Will Transform How We Live, Work, and Think by Viktor Mayer-Schönberger and Kenneth Cukier
2. Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking by Foster Provost and Tom Fawcett

Course Code	MCE-E108
Course Name	Machine Learning
Credits	3
Pre-Requisites	Mathematics and AI

Total Number of Lectures : 35

COURSE OBJECTIVE

- To learn the concept of how to learn patterns and concepts from data without being explicitly programmed in various IOT nodes.
- To design and analyse various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

LECTURE WITH BREAKUP	NO. OF LECTURES
UNIT-I **Supervised Learning (Regression/Classification)** • Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes. • Linear models: Linear Regression, Logistic Regression, Generalized Linear Models • Support Vector Machines, Nonlinearity and Kernel Methods • Beyond Binary Classification: Multi-class/Structured Outputs, Ranking	6
UNIT-II **Unsupervised Learning** • Clustering: K-means/Kernel K-means • Dimensionality Reduction: PCA and kernel PCA • Matrix Factorization and Matrix Completion • Generative Models (mixture models and latent factor models)	5
UNIT-III Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests)	6
UNIT-IV Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning	7
UNIT-V Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference.	6
UNIT-VI Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.	5

COURSE OUTCOMES

After completion of course, students would be able to:

- Extract features that can be used for a particular machine learning approach in various IOT applications.

- To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach.
- To mathematically analyse various machine learning approaches and paradigms. Students will be able to explain the key elements of robot cell design and identify potential applications of robots in various industries

References:

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Course Code	MCE-A104
Course Name	Research Methodology and IPR
Credits	3
Pre-Requisites	

Total Number of Lectures : 35

COURSE OBJECTIVE
• Understand research problem formulation. • Analyze research related information. • Follow research ethics • Understand that today's world is controlled by Computer, Information Technology, but tomorrow's world will be ruled by ideas, concepts, and creativity. • Understanding that when IPR would take such an important place in the growth of an individual nation, it is needless to emphasize the need for information about Intellectual Property rights to be promoted among students in general & engineering in particular. • Understanding that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about economic growth and social benefits.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations	7
Unit 2 Effective literature studies approaches, analysis, Plagiarism, Research ethics,	6
Unit 3 Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee	7
Unit 4 Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.	6
Unit 5 Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications.	5
Unit 6 New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.	4

COURSE OUTCOMES
After completion of course, students would be able to: • You will gain the ability to formulate well-defined research questions, identify relevant research methodologies (quantitative, qualitative, mixed methods), and design effective research plans.

- You will learn to identify potential biases in research and develop strategies for ensuring the objectivity and validity of your own research findings.
- You will develop skills in creating presentations to present your research in a clear and engaging manner.

References:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"
3. Ranjit Kumar, 2 nd Edition , "Research Methodology: A Step by Step Guide for beginners"
4. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007.
5. Mayall , "Industrial Design", McGraw Hill, 1992.
6. Niebel , "Product Design", McGraw Hill, 1974.
7. Asimov , "Introduction to Design", Prentice Hall, 1962.
8. Robert P. Merges, Peter S. Menell, Mark A. Lemley, " Intellectual Property in New Technological Age", 2016.
9. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

Year I - Semester II

Course Code	MCE-C201
Course Name	Advance Algorithms
Credits	3
Pre-Requisites	Basic data structures and algorithms

Total Number of Lectures:35

COURSE OBJECTIVE
• Introduce students to the advanced methods of designing and analyzing algorithms. • The student should be able to choose appropriate algorithms and use it for a specific problem. • To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems. • Students should be able to understand different classes of problems concerning their computation difficulties.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 **Sorting:** Review of various sorting algorithms, topological sorting. **Graph:** Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.	6
Unit 2 **Matroids:** Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. **Graph Matching:** Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.	5
Unit 3 **Flow-Networks:** Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. **Matrix Computations:** Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.	6
Unit 4 **Shortest Path in Graphs:** Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. **Modulo Representation of integers/polynomials:** Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. **Discrete Fourier Transform (DFT):** In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm	6
Unit 5 **Linear Programming:** Geometry of the feasibility region and Simplex algorithm, **NP-completeness:** Examples, proof of NP-hardness and NP-completeness. **One or more of the following topics based on time and interest:** Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm	7
Unit 6 Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.	5

COURSE OUTCOMES
After completion of course, students would be able to: ● Analyze the complexity/performance of different algorithms. ● Determine the appropriate data structure for solving a particular set of problems. ● Categorize the different problems in various classes according to their complexity. ● Students should have an insight of recent activities in the field of advanced data structure.

References:

1. Introduction to Algorithms" byCormen, Leiserson, Rivest, Stein.
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.
3. "Algorithm Design" by Kleinberg and Tardos.

Course Code	MCE-C202
Course Name	Soft Computing
Credits	3
Pre-Requisites	Mathematics

Total Number of Lectures : 35

COURSE OBJECTIVE
• To introduce soft computing concepts and techniques and foster their abilities in designing appropriate techniques for a given scenario. • To implement soft computing based solutions for real-world problems. • To give students knowledge of non-traditional technologies and fundamentals of artificial neural networks, fuzzy sets, fuzzy logic, genetic algorithms. • To provide students with hand-on experience on MATLAB to implement various strategies.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to Soft Computing and Neural Networks:** Evolution of Computing: Soft Computing Constituents, From Conventional AI to Computational Intelligence : Machine Learning Basics.	7
Unit 2 **Fuzzy Logic:** Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations,Membership Functions: Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making.	6
Neural Networks: Machine Learning Using Neural Network, Adaptive Networks, Feed forward Networks, Supervised Learning Neural Networks, Radial Basis Function Networks : Reinforcement Learning, Unsupervised Learning Neural Networks, Adaptive Resonance architectures, Advances in Neural networks.	7
Unit 4 **Genetic Algorithms:** Introduction to Genetic Algorithms (GA), Applications of GA in Machine Learning : Machine Learning Approach to Knowledge Acquisition.	4
Unit 5 **Matlab/Python Lib:** Introduction to Matlab / Python, Arrays and array operations, Functions and Files, Study of neural network toolbox and fuzzy logic toolbox, Simple implementation of Artificial Neural Network and Fuzzy Logic.	7
Unit 6 Recent Trends in deep learning, various classifiers, neural networks and genetic algorithms. Implementation of recently proposed soft computing techniques.	4

COURSE OUTCOMES

After completion of course, students would be able to:

- Identify and describe soft computing techniques and their roles in building intelligent machines.
- Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.
- Apply genetic algorithms to combinatorial optimization problems.
- Evaluate and compare solutions by various soft computing approaches for a given problem.

References:

1. Jyh:Shing Roger Jang, Chuen:Tsai Sun, EijiMizutani, Neuro:Fuzzy and Soft Computing , Prentice:Hall of India, 2003.
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic:Theory and Applications , Prentice Hall, 1995.
3. MATLAB Toolkit Manual

Elective 2

Course Code	MCE-E201
Course Name	Information and Network Security
Credits	3
Pre-Requisites	Computer Networks and web Development

Total Number of Lectures : 35

COURSE OBJECTIVE
• Provide a comprehensive understanding of core information security principles, including the CIA triad (Confidentiality, Integrity, Availability) and information security domains. • Introduce students to fundamental cryptographic concepts • Explore various network security concepts, protocols (SSL/TLS, IPsec, VPNs), and best practices for securing network infrastructure. • Foster the ability to develop and implement security policies, procedures, and compliance measures within an organization.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to Information Security:** Information Security Fundamentals: CIA triad (Confidentiality, Integrity, Availability), security domains (physical, network, application, data). Security Threats and Attacks: Types of threats (interception, interruption, modification), common attack vectors (social engineering, malware). Security Risk Management: Risk identification, assessment, mitigation, and acceptance. Cryptography Fundamentals: Symmetric and asymmetric cryptography, hashing algorithms, digital signatures.	6
Unit 2 **Network Security:** Network Security Architecture: Network security models (DMZ, firewalls), secure network design principles. Network Security Protocols: Secure Socket Layer (SSL)/Transport Layer Security (TLS), IPsec, VPNs. Wireless Network Security: Wireless security standards (WEP, WPA, WPA2), vulnerabilities and countermeasures. Intrusion Detection and Prevention Systems (IDS/IPS): Detection methods, signature-based vs. anomaly-based IDS, intrusion prevention techniques	5
Unit 3 **System Security:** Operating System Security: Hardening operating systems (Windows, Linux), user access control, privilege escalation. Application Security: Secure software development practices (OWASP Top 10), code vulnerabilities, static and dynamic analysis tools. Malware Analysis: Types of malware (viruses, worms, trojans), analysis techniques, signature-based and behavior-based detection. Digital Forensics and Incident Response: Digital evidence collection, forensic analysis methods, incident response procedures.	7
Unit 4 **Advanced Information Security Topics:** Cloud Security: Security considerations for cloud computing environments (IaaS, PaaS, SaaS), shared responsibility model, cloud security standards. IoT Security: Threats and vulnerabilities in IoT devices, secure communication protocols, secure development practices for IoT systems. Blockchain Security: Security considerations and potential vulnerabilities in blockchain technology, securing smart contracts. Privacy-Enhancing Technologies: Anonymization, pseudonymization, differential privacy for protecting user data.	6
Unit 5 **Security Management and Standards:** Security Policies and Procedures: Developing organization-wide security policies, procedures, and awareness programs. Security Standards	7

and Regulations: Compliance with security standards (e.g., PCI DSS, HIPAA), legal aspects of information security. Security Audits and Penetration Testing: Methodology for security audits, penetration testing techniques, ethical hacking. Emerging Security Trends: Exploring cutting-edge security threats and technologies (e.g., artificial intelligence security, quantum computing security).	
Unit 6 Other topics: Biometric authentication, Secure E-Commerce (ex. SET), Smart Cards, Security in Wireless Communication. recent trends in IOT security, IDS and Biometric.	4

COURSE OUTCOMES

After completion of course, students would be able to:

- Students will be able to explain the CIA triad, identify information security domains (physical, network, application, data), and analyze security threats and attack vectors.
- Students will be able to describe network security models (DMZ, firewalls), explain the functionalities of secure network protocols (SSL/TLS, IPsec, VPNs), and analyze wireless network security considerations.
- Students will be able to differentiate between various malware types (viruses, worms, trojans), explain malware analysis techniques, and discuss digital forensics and incident response procedures.
- Students will be able to contribute to developing security policies and procedures, understand legal aspects of information security, and explain the role of security audits and penetration testing.

References:

1. W. R. Cheswick and S. M. Bellovin. Firewalls and Internet Security. Addison Wesley, 1994.
2. W. Stallings. Cryptography and Network Security. Prentice Hall, 1999.
3. B. Schneier. Applied Cryptography. Wiley, 1999.

Course Code	MCE-E202
Course Name	Ethical Hacking
Credits	3
Pre-Requisites	Basics of Computer Networks and security

Total Number of Lectures : 35

COURSE OBJECTIVE

- Introduces the concepts of Ethical Hacking and gives the students the opportunity to learn about different tools and techniques in Ethical hacking and security and practically apply some of the tools.
- Foster the development of a strong ethical code of conduct for ethical hackers, emphasizing professionalism, responsible disclosure, and adherence to legal boundaries.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 Introduction to Ethical Disclosure: Ethics of Ethical Hacking, Ethical Hacking and the legal system, Proper and Ethical Disclosure.	6
Unit 2 Penetration Testing and Tools: Using Metasploit, Using BackTrackLiveCD Linux Distribution.	7
Unit 3 Vulnerability Analysis: Passive Analysis, Advanced Static Analysis with IDA Pro, Advanced Reverse Engineering	6
Unit 4 Client-side browser exploits, Exploiting Windows Access Control Model for Local Elevation Privilege, Intelligent Fuzzing with Sulley, From Vulnerability to Exploit	7
Unit 5 Malware Analysis: Collecting Malware and Initial Analysis, Hacking Malware	5
Unit 6 Case study of vulnerability of cloud platforms and mobile platforms & devices.	4

COURSE OUTCOMES

After completion of course, students would be able to:

- Understand the core concepts related to malware, hardware and software vulnerabilities and their causes.
- Understand ethics behind hacking and vulnerability disclosure.
- Appreciate the Cyber Laws and impact of hacking.
- Exploit the vulnerabilities related to computer system and networks using state of the art tools and technologies

References:

1. Shon Harris, Allen Harper, Chris Eagle and Jonathan Ness, Gray Hat Hacking: The Ethical Hackers' Handbook, TMH Edition
2. Jon Erickson, Hacking: The Art of Exploitation, SPD

Course Code	MCE-E203
Course Name	Parallel Computing
Credits	3
Pre-Requisites	Computer Architecture

Total Number of Lectures : 35

COURSE OBJECTIVE
• Classify parallel architecture parameters that are essential for the classification of modern parallel processing systems. • Describe the methodologies employed for synchronization and memory consistency and cache coherence in shared memory systems. • Describe and compare the different types of interconnects employed in parallel processing systems. • Describe how the performance of a parallel system can be measured, list possible sources for performance losses and propose ways to improve the performance of a system.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit1 Introduction to Parallel Computing Architectures, parallel hardware/multi-cores, Processes and threads, Programming models: shared memory and message passing, Amdahl's Law.	5
Unit2 Introduction to parallel hardware: Multi-cores and multiprocessors, shared memory and message passing architectures, cache hierarchy and coherence, sequential consistency.	7
Unit3 Introduction to parallel software: Steps involved in developing a parallel program, Dependence analysis, Domain decomposition, Task assignment: static and dynamic, Performance issues: 4C cache misses, inherent and artifactual communication, false sharing, computation-to-communication ratio as a guiding metric for decomposition, hot spots and staggered communication.	6
Unit4 Shared memory parallel programming: Synchronization Locks and barriers, Hardware primitives for efficient lock implementation, Lock algorithms, Relaxed consistency models, High-level language memory models (such Java and/or C++), Memory fences. Developing parallel programs with UNIX fork model: IPC with shared memory and message passing, UNIX semaphore and its all-or-none semantic. Developing parallel programs with POSIX thread library, Thread creation, Thread join, Mutex, Condition variables. Developing parallel programs with OpenMP directives: Parallel for, Parallel section, Static, dynamic, guided, and runtime scheduling, Critical sections and atomic operations, Barriers Reduction.	8
Unit 5 Introduction to GPU programming: GPU architecture, Introduction to CUDA programming, Concept of SIMD and SIMT computation, Thread blocks, Warps, Global memory, Shared memory, Thread divergence in control transfer.	5
Unit 6 Recent trends in Parallel Programming Models and Paradigms. Case study of parallel hardware which include shared memory architecture and message passing architectures for efficient computing.	4

COURSE OUTCOMES
After completion of course, students would be able to: ● Understand the methodologies employed for synchronization and memory consistency and cache coherence in shared memory systems.

References:

1. Peter S Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann,2011.
2. M Herlihy and N Shavit, The Art of Multiprocessor Programming Morgan Kaufmann, 2008.
3. JL Hennessy and DA Patterson, Computer Architecture: A Quantitative Approach,4th Ed., Morgan Kaufmann/Els India, 2006.

Course Code	MCE-E204
Course Name	Internet of Things
Credits	3
Pre-Requisites	Basic electronics and networking

Total Number of Lectures : 35

COURSE OBJECTIVE
• Able to understand the application areas of IOT. • Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks. • Able to understand building blocks of Internet of Things and characteristics

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to the Internet of Things:** What is IoT? History, evolution, and key characteristics of IoT. Applications of IoT: Exploring diverse applications across various industries (e.g., smart cities, healthcare, agriculture). Building Blocks of IoT Systems: Sensors, actuators, microcontrollers, communication modules. IoT Architectures: Layered architectures (perception, network, application, management)	6
Unit 2 **Hardware Platforms and Communication Technologies:** Popular IoT Development Boards: Introduction to platforms like Arduino, Raspberry Pi, and ESP8266. Sensors and Actuators: Types of sensors (temperature, humidity, motion) and actuators (relays, motors, LEDs). Interfacing Sensors and Actuators: Connecting sensors and actuators to microcontroller boards. Wireless Communication Protocols: Wi-Fi, Bluetooth, Zigbee, LoRaWAN for IoT communication	8
Unit 3 **Data Acquisition, Processing, and Analytics :** Data Acquisition Strategies: Techniques for collecting data from sensors in IoT devices. Data Storage and Management: Cloud platforms and databases for storing and managing IoT data. Data Processing Techniques: Stream processing, time-series analysis, data visualization for IoT data. Machine Learning for IoT: Introduction to using machine learning for anomaly detection and predictive maintenance in IoT applications.	6
Unit 4 **Security and Privacy in IoT:** Security Challenges in IoT: Vulnerability to cyberattacks, data breaches, and privacy concerns. Authentication and Authorization: Implementing secure access control mechanisms for IoT devices. Data Security and Encryption: Techniques for securing data at rest, in transit, and in use. Security Best Practices: Guidelines for secure development, deployment, and management of IoT devices.	6
Unit 5 Interface Electronic Circuit for Smart Sensors and Challenges forInterfacing the Smart Sensor, Usefulness of Silicon Technology in Smart Sensor And Future scope of research in smart sensor	5
Unit 6 **Advanced Topics and Future of IoT:** IoT Cloud Platforms: Exploring popular cloud platforms for developing and managing IoT applications. Edge Computing for IoT: Decentralized processing and decision-making at the edge of the network. Artificial Intelligence for IoT: Exploring the integration of AI techniques for advanced IoT functionalities. Emerging Trends in IoT: Discussing recent advancements and the future direction of IoT technology.	4

COURSE OUTCOMES
After completion of course, students would be able to: ● Understand the vision of IoT from a global context. ● Determine the Market perspective of IoT. ● Use of Devices, Gateways and Data Management in IoT. ● Application of IoT in Industrial and Commercial Building Automation and Real World Design Constraints. ● Building state of the art architecture in IoT.

References:

1. Yasuura, H., Kyung, C.-M., Liu, Y., Lin, Y.-L., Smart Sensors at the IoT Frontier, Springer International Publishing
2. Kyung, C.-M., Yasuura, H., Liu, Y., Lin, Y.-L., Smart Sensors and Systems, Springer International Publishing

Course Code	MCE-E205
Course Name	Advance Artificial Intelligence
Credits	3
Pre-Requisites	Mathematics and Machine Learning Foundations

Total Number of Lectures : 35

COURSE OBJECTIVE

- To equip students with a comprehensive understanding of advanced Deep Learning architectures and their applications.
- Introduce students to the fundamental concepts and algorithms of Reinforcement Learning and its applications
- Deepen understanding of Natural Language Processing (NLP) techniques and explore applications like machine translation and sentiment analysis.
- To expose students to cutting-edge topics in AI including Explainable AI, Evolutionary Computation, and Artificial General Intelligence.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 Deep Neural Networks (DNNs): Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks. Autoencoders and Variational Autoencoders (VAEs): Unsupervised learning for dimensionality reduction and data generation. Generative Adversarial Networks (GANs): Generative models for creating realistic data (images, text). Deep Learning Optimization Techniques: Gradient descent variants, Adam, RMSprop, regularization techniques.	5
Unit 2 Reinforcement Learning (RL) Fundamentals: Markov Decision Processes (MDPs), Q-learning, Policy Gradients. Deep Reinforcement Learning (DRL): Deep Q-Networks (DQNs), Policy Gradient with Actor-Critic methods. Multi-Agent Reinforcement Learning: Challenges and cooperative/competitive learning algorithms. Applications of Reinforcement Learning: Robotics control, game playing (AlphaGo, OpenAI Gym).	5
Unit 3 Natural Language Processing (NLP) Fundamentals: Text preprocessing, tokenization, stemming, lemmatization. Word Embeddings and Language Models: Word2Vec, GloVe, Recurrent Neural Networks for language modeling. Machine Translation and Text Summarization: Neural Machine Translation (NMT), encoder-decoder architectures. Sentiment Analysis and Text Classification: Convolutional Neural Networks (CNNs) for text analysis.	7
Unit 4 Computer Vision with Deep Learning: Object detection (YOLO, R-CNN), image segmentation (U-Net). Explainable Artificial Intelligence (XAI): Techniques for understanding and interpreting AI models.	7
Unit 5 Evolutionary Computation and Genetic Algorithms: Population-based optimization techniques inspired by nature. Artificial General Intelligence (AGI): Challenges and potential approaches for achieving human-level intelligence.	6
Unit 6 Research Paper Presentations: Students present research papers on cutting-edge AI topics. AI Project Development: Students design, implement, and evaluate an advanced AI project in a chosen domain (e.g., healthcare, robotics, finance). Ethical Considerations in AI: Bias in AI	5

systems, fairness, transparency, and responsible AI development. The Future of AI: Potential impact of AI on society and industry, responsible AI development practices.	

COURSE OUTCOMES

- Students will be able to design and implement Deep Learning architectures (CNNs, RNNs, LSTMs) for various tasks like image recognition, natural language processing, and time series forecasting.
- Students will be able to understand and apply Reinforcement Learning algorithms (Q-learning, DQN) for agent training in complex decision-making scenarios.
- Students will be able to design, implement, and evaluate an advanced AI project in a chosen domain, considering ethical considerations and responsible AI development practices.
- Students will be able to critically evaluate research papers on advanced AI topics

References:

- Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
- Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
- Christopher M. Bishop, Pattern Recognition and Machine Learning.
- Deep Learning, by Goodfellow, Bengio, and Courville.

Course Code	MCE-E206
Course Name	Genetic Algorithms
Credits	3
Pre-Requisites	Basic Mathematics and problem solving

Total Number of Lectures : 35

COURSE OBJECTIVE
• Provide a comprehensive understanding of the limitations of traditional optimization methods and introduce Genetic Algorithms as an alternative approach. • Explain the core concepts of GAs, including selection, crossover, mutation, and schema processing. • Equip students with the knowledge to implement a basic GA using appropriate data structures and operators (reproduction, crossover, mutation). • Introduce advanced techniques for genetic search, including dominance, inversion operators, niching, and multiobjective optimization • Expose students to real-world applications of GAs in various engineering problems, including job-shop scheduling and routing.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
UNIT-1 Introduction: Robustness of Traditional Optimization and Search Methods, The goals of Optimization, How are Genetic Algorithms Different from Traditional Methods?, A simple genetic algorithm, Genetic algorithms at work – a simulation by hand, Grist for the Search Mill – important similarities, Similarity templates (Schemata), Learning the Lingo.	8
Unit-2 Genetic Algorithms Revisited: Mathematical Foundations – Who shall live and who shall die? The fundamental theorem, Schema Processing at work: An example by hand revisited, the two armed and K-armed bandit problem, How many schemata are processed usefully?, The building block hypothesis, another perspective: the minimal Deceptive problem, Schemata revisit: similarity templates as hyper planes.	7
Unit-3 Computer Implementation Of A Genetic Algorithm – Data structures, reproduction, crossover, and mutation, A time to reproduce, a time to cross, get with the main program, How well does it work? Mapping objective functions to fitness form, fitness scaling, codings, a multiparameter, mapped, fixed-point coding, discretization, Constraint Handling	7
Unit-4 Techniques In Genetic Search – Dominance, diploidy and abeyance, inversion and other reordering operators, other micro operators, niche and speciation, multiobjective optimization - Knowledge-based techniques, genetic algorithms and parallel processors.	5
Unit-5 Multi objective evolutionary optimization: Pareto optimality, multi-objective evolutionary algorithms: MOGA, NSGA-II, etc. Applications of GA in engineering problems, job-shop scheduling and routing problems	4
Unit-6 Future Directions and Societal Impact of GAs: Challenges of GAs in Dynamic Environments, Hybridization of GAs with Other Optimization Techniques, Ethical Considerations in GA Design and Use, Societal Impact of GAs and Future Trends	4

COURSE OUTCOMES
After completion of course, students would be able to: ● Students will be able to differentiate between the limitations of traditional optimization methods and the strengths of Genetic Algorithms. ● Students will be able to articulate the concept of schemata and its role in the search process of GAs. ● Students will be able to implement a basic GA in a programming language using appropriate data structures and operators for reproduction, crossover, and mutation. ● Students will be able to discuss advanced genetic search techniques such as dominance, inversion, niching, and multi objective optimization ● Students will be able to identify potential applications of GAs in solving engineering problems like job-shop scheduling and routing.

References:

1. David E.Goldberg, "Genetic Algorithms" – 1/e, Pearson Education.
2. Genetic algorithms in search, optimization and Machine learning, By David E. Gold Berg Pearson Edition
3. An Introduction to Genetic Algorithm by Melanie Mitchell
4. The Simple Genetic Algorithm Foundation & Theories by Michael P. Vosk

Elective 3

Course Code	MCE-E114 / MCE-E213
Course Name	Cloud Computing
Credits	3
Pre-Requisites	

Total Number of Lectures:35

COURSE OBJECTIVE
• The student will also learn how to apply trust-based security model to real-world security problems. • An overview of the concepts, processes, and best practices needed to successfully secure information within Cloud infrastructures. • Students will learn the basic Cloud types and delivery models and develop an understanding of the risk and compliance responsibilities and Challenges for each Cloud type and service delivery model.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
Unit 1 **Introduction to Cloud Computing:** Online Social Networks and Applications, Cloud introduction and overview, Different clouds, Risks, Novel applications of cloud computing	4
Unit 2 **Cloud Computing Architecture:** Requirements, Introduction Cloud computing architecture, On Demand Computing Virtualization at the infrastructure level,Security in Cloud computing environments, CPU Virtualization, A discussion on Hypervisors Storage Virtualization Cloud Computing Defined, The SPI Framework for Cloud Computing, The Traditional Software Model, The Cloud Services Delivery Model **Cloud Deployment Models** Key Drivers to Adopting the Cloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise	8
Unit 3 **Security Issues in Cloud Computing:** Infrastructure Security, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security **Identity and Access Management,** Trust Boundaries and IAM, IAM Challenges, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud, Cloud Authorization Management	6
Unit 4 **Security Management in the Cloud:** Security Management Standards, Security Management in the Cloud, Availability Management: SaaS, PaaS, IaaS. **Privacy Issues:** Privacy Issues, Data Life Cycle, Key Privacy Concerns in the Cloud, Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications, U.S. Laws and Regulations, International Laws and Regulations	7
Unit 5: **Audit and Compliance:** Internal Policy Compliance, Governance, Risk, and Compliance (GRC), Regulatory/External Compliance, Cloud Security Alliance, Auditing the Cloud for Compliance, Security-as-a-Cloud	6
Unit 6 **ADVANCED TOPICS:** Recent developments in hybrid cloud and cloud security .	4

COURSE OUTCOMES
After completion of course, students would be able to:

- Identify security aspects of each cloud model
- Develop a risk-management strategy for moving to the Cloud
- Implement a public cloud instance using a public cloud service provider
- Apply trust-based security model to different layer

References:

1. Cloud Computing Explained: Implementation Handbook for Enterprises, John Rhoton, Publication Date: November 2, 2009
2. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), Tim Mather, ISBN-10: 0596802765, O'Reilly Media, September 2009

Course Code	MCE-E109 / MCE-E208
Course Name	Software testing and auditing
Credits	3
Pre-Requisites	Basic Software Engineering

Total Number of Lectures : 35

COURSE OBJECTIVE

- Provide a comprehensive understanding of testing as an engineering activity and its role in ensuring software quality.
- Equip students with essential test case design strategies, including black-box techniques (equivalence partitioning, boundary value analysis) and white-box techniques (code coverage, path testing).
- Explore the different levels of software testing (unit, integration, system, regression) and their significance in the testing process.
- Expose students to software audit principles and differentiate them from other review practices.

LECTURE WITH BREAKUP	NO. OF LECTURES
UNIT-1 Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control.	5
UNIT-2 **TESTING BASICS:** Testing as an engineering activity – Role of process in software quality – Testing as a process – Basic definitions – Software testing principles – The tester's role in a software development organization – Origins of defects – Defect classes – The defect repository and test design – Defect examples – Developer / Tester support for developing a defect repository.	6
UNIT-3 **TEST CASE DESIGN:** Introduction to testing design strategies – The smarter tester – Test case design strategies – Using black box approach to test case design – Random testing – Equivalence class partitioning – Boundary value analysis – Other black box test design approaches – Black box testing and COTS – Using white box approach to test design – Test adequacy criteria – Coverage and control flow graphs – Covering code logic – Paths – Their role in white box based test design – Additional white box test design approaches – Evaluating test adequacy criteria.	7
UNIT-4 **LEVELS OF TESTING:** The need for levels of testing – Unit test – Unit test planning – Designing the unit tests – The class as a testable unit – The test harness – Running the unit tests and recording results – Integration tests – Designing integration tests – Integration test planning – System test – The different types – Regression testing – Alpha, beta and acceptance tests.	6
UNIT-5 **TEST MANAGEMENT -CONTROLLING AND MONITORING:**Basic concepts – Testing, debugging goals, policies – Test planning – Test plan components – Test plan attachments – Locating test items – Reporting test results – The role of three groups in test planning and policy development – Process and the engineering disciplines – Introducing the test specialist – Skills needed by a test specialist – Building a testing group. Defining terms – Measurements and milestones for controlling and monitoring – Status meetings – Reports and control issues – Criteria for test completion – SCM – Types of reviews – Developing a review program – Components of review plans – Reporting review results.	7
UNIT-6: AUDITING Software audit review, software audits Vs software peer reviews and software management reviews. Objectives and participants Initiator, Lead Auditor, Recorder, Auditors, Audited Organization.	4

COURSE OUTCOMES
After completion of course, students would be able to: • Students will be able to explain the importance of testing in software development and articulate the principles of effective testing. • Students will be able to use various black-box and white-box test case design techniques to create comprehensive test cases for software applications. • Students will be able to develop test plans, manage and monitor the testing process, and report test results effectively. • Students will be able to understand the concept of software audits and distinguish them from other software review methods.

References:

1. SrinivasanDesikan, Gopalaswamy Ramesh, “Software Testing: Principles and Practices”, Pearson 2012
2. Aditya P. Mathur, “Foundations of Software Testing”, Pearson, 2008
3. Paul Ammann, Jeff Offutt, “Introduction to Software Testing”, Cambridge University Press, 2008
4. Paul C. Jorgensen, “Software Testing: A Craftsman's Approach”, Auerbach Publications, 2008
5. IEEE Standard for Software Reviews, clause 3.2

Course Code	MCE-E110 / MCE-E209
Course Name	Mobile Application Development
Credits	3
Pre-Requisites	Basic Programming and OOPs concepts

Total Number of Lectures : 35

COURSE OBJECTIVE

- Introduce students to the mobile app ecosystem, different types of mobile applications, and popular mobile operating systems.
- Equip students with the skills to develop native Android applications using Android Studio and the Java/Kotlin programming languages.
- Guide students in building native iOS applications using Xcode and the Swift programming language.
- Expose students to advanced mobile development topics like API integration, real-time features, and app deployment strategies.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 **Introduction to Mobile Application Development:** Mobile App Ecosystem: (Types of mobile applications (native, web, hybrid), Popular mobile operating systems (Android, iOS), Mobile app development lifecycle (planning, design, development, testing, deployment), Introduction to User Interface (UI) and User Experience (UX) Design: Design principles for mobile applications, User-centered design methodologies	6
Unit 2 **Native App Development – Android:** Introduction to Android Development Environment: (Setting up Android Studio, Understanding Android application components (activities, services, broadcast receivers), Building User Interfaces with Android: Layouts, views, widgets, and UI design tools, Implementing event handling and user interactions Android Development Fundamentals: Working with Activities, Intents, and Fragments, Data storage options (Shared Preferences, SQLite)	7
Unit 3 **Native App Development – iOS:** Introduction to iOS Development Environment: (Setting up Xcode and Swift programming language, Understanding iOS application architecture (MVC, MVVM), Building User Interfaces with iOS: Storyboards and SwiftUI for creating UIs, Auto Layout and constraints for responsive design, iOS Development Fundamentals: Working with View Controllers, Views, and Navigation, Data storage options (Core Data, NSUserDefaults)	5
Unit 4 **Cross-Platform Development:** Introduction to Cross-Platform Frameworks: Benefits and limitations of cross-platform development Introduction to Flutter and React Native, Building Cross-Platform Apps with Flutter (or React Native): Hands-on experience with chosen cross-platform framework, Creating UIs, handling user interactions, and integrating APIs, Advantages and Use Cases for Cross-Platform Development: When to choose native vs. cross-platform development, Building cross-platform apps for different target platforms.	6
Unit 5 **Advanced Topics and Deployment:** API Integration: Consuming APIs for data retrieval and functionality, Authentication and authorization techniques; Real-time Features and Push Notifications: Implementing real-time updates and notifications in mobile apps; Deployment and App Stores: App signing, publishing process, and app store guidelines Testing and quality assurance for mobile applications.	6

Unit 6 **Alternative Mobile Development Approaches:** Hybrid Mobile App Development and Game Development for Mobile Platforms:	5
COURSE OUTCOMES	
After completion of course, students would be able to: ● Students will be able to explain the different types of mobile applications, popular mobile operating systems, and the mobile app development life cycle. ● Students will be able to set up the Android development environment, understand core Android components, and build basic Android applications. ● Students will be able to navigate the iOS development environment, understand iOS application architecture, and create native iOS apps using Swift ● Students will be able to leverage cross-platform development frameworks to build mobile apps that work across different platforms. ● Students will be able to integrate APIs into mobile applications, implement real-time features, and prepare their apps for deployment on app stores	

References:

1. Head First Mobile Development: A Brain-Friendly Guide to Building Apps for iOS and Android by Becky Fuller
2. Android Programming: The Big Nerd Ranch Guide by Chris Stewart and Kristin Marsicano

Course Code	MCE-E111 / MCE-E210
Course Name	Scientific Computing with Python
Credits	3
Pre-Requisites	Mathematics and Basic Programming

Total Number of Lectures : 35

COURSE OBJECTIVE

- Equip students with the ability to set up a scientific computing environment using Python and popular scientific libraries (NumPy, SciPy, Matplotlib).
- Introduce core concepts of numerical linear algebra and demonstrate their implementation using NumPy for solving linear systems of equations and performing matrix decompositions.
- Explore various numerical calculus techniques, including numerical differentiation, integration, and optimization algorithms.
- Equip students with the skills to create informative and visually appealing scientific visualizations using Matplotlib.
- Expose students to advanced topics like solving ordinary and partial differential equations and showcase real-world applications of scientific computing with Python through case studies.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit 1 **Basics of Python:** Why Python?, Basics: Variables, Data Types, Operators, control flow and functions.	7
Unit 2 **Introduction to Scientific Computing and Python:** Introduction to Scientific Computing: Applications in various scientific disciplines (physics, chemistry, engineering) Numerical vs. symbolic computation Python for Scientific Computing: Setting up a scientific computing environment (Python, libraries), Introduction to NumPy arrays and data structures, Basic programming concepts (variables, data types, control flow)	6
Unit 3 **Numerical Linear Algebra:** Linear Algebra Fundamentals (Review): Matrices, vectors, determinants, eigenvalues, eigenvectors NumPy for Linear Algebra: (Linear algebra operations with NumPy arrays, Solving linear systems of equations (LU decomposition, Gaussian elimination), Matrix decompositions (eigenvalue decomposition, singular value decomposition)	6
Unit 4 **Numerical Calculus:** Numerical Differentiation: (Finite difference methods for differentiation, Numerical errors and error analysis), Numerical Integration: (Trapezoidal rule, Simpson's rule, Gaussian quadrature, Numerical integration for definite and indefinite integrals), Optimization Techniques: (Gradient descent, root-finding algorithms (Newton-Raphson method)	5
Unit 5 **Data Analysis and Visualization with SciPy and Matplotlib:** Introduction to SciPy: (Special functions, statistical functions, optimization routines, Data Analysis with SciPy: (Data loading and manipulation (CSV, text files), Descriptive statistics, hypothesis testing), Scientific Visualization with Matplotlib: (Creating various plots (line plots, scatter plots, histograms), Customization of plots for effective data visualization	6
Unit 6: **Advanced Topics and Applications:** Solving Ordinary Differential Equations (ODEs): Euler's method, Runge-Kutta methods. Introduction to Partial Differential Equations (PDEs): (Finite difference methods for solving simple PDEs), Applications of Scientific Computing with Python (case studies): (Examples from chosen scientific disciplines).	5

COURSE OUTCOMES
After completion of course, students would be able to: ● Students will be able to explain the role of scientific computing in various scientific disciplines and differentiate between numerical and symbolic computation. ● Students will be able to use SciPy for data analysis tasks, including data loading, manipulation, and statistical analysis ● Students will be able to choose appropriate numerical calculus methods based on the specific problem and desired level of accuracy. ● Students will be able to understand basic concepts of solving partial differential equations using finite difference methods. ● Students will be able to present their scientific findings and problem-solving approaches using scientific computing methods.

References:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Müller and Sarah Guido, O'Reilly, 2016.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Aurelien Geron, O'Reilly, 2017.

Course Code	MCE-E112 / MCE-E211
Course Name	Quantum computing
Credits	3
Pre-Requisites	Linear algebra and probability

Total Number of Lectures : 35

COURSE OBJECTIVE

- Introduce the fundamental building blocks of quantum computation.
- Provide a foundation in the mathematical and physical principles underlying quantum computation.
- Explore the concept of quantum circuits and equip students with design skills.
- Compare classical and quantum information theory, introduce groundbreaking applications.
- Investigate the potential of quantum algorithms, analyze their relationship to classical complexity.
- Discuss the challenges of noise and error correction in quantum computation.

LECTURE WITH BREAKUP	NO. OF LECTURES
Unit I **Qubits and Quantum Mechanics:** Introduction to qubits and their differences from classical bits, Bloch sphere representation of qubit states, Multi-qubit systems and their exponential power, Introduction to Hilbert spaces.	7
Unit II **Mathematical Foundations:** Probability and measurements in quantum computation, Entanglement and its significance, Density operators for representing mixed quantum states, Basic principles of quantum mechanics relevant to computation, Limitations of measuring qubits in bases other than the computational basis.	6
Unit III **Quantum Circuits:**Single-qubit gates used in quantum circuits, Multi-qubit gates for manipulating quantum information, Designing quantum circuits for specific computational tasks.	6
Unit IV **Quantum Information and Cryptography:** Comparison of classical and quantum information theory, Bell states and quantum nonlocality, Quantum teleportation process and its implications, Principles of quantum cryptography and the no-cloning theorem.	5
Unit V **Quantum Algorithms:** Implementing classical algorithms on quantum computers, Relationship between quantum and classical complexity classes (e.g., P vs. BQP), Principles and applications of key quantum algorithms (Deutsch's algorithm, Deutsch-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm).	6
Unit VI **Noise and Error Correction:** Challenges of noise and decoherence in quantum computation, Graph states and quantum error correction codes, Principles of fault-tolerant quantum computation and its importance.	5

COURSE OUTCOMES
After completion of course, students would be able to: ● Students will be able to define qubits and explain their differences from classical bits. ● Students will be able to apply probability concepts to analyze quantum measurements and outcomes. ● Students will be able to identify and describe various single-qubit gates used in quantum circuits ● Students will be able to compare and contrast classical and quantum information theory. ● Students will be able to explain how classical algorithms can be implemented on quantum computers.

References:

1. Nielsen M. A., Quantum Computation and Quantum Information,

   Cambridge University Press. 2002
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific. 2004
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms, 2000

Course Code	MCE-E113 / MCE-E212
Course Name	Robotics
Credits	3
Pre-Requisites	Mathematics and AI

Total Number of Lectures : 35

COURSE OBJECTIVE

- Introduce students to the definition, history, and terminology of robotics, emphasizing the Laws of Robotics.
- Explore various end effectors used in robots, including grippers (mechanical, magnetic, vacuum, air operated) and their design considerations.
- Introduce robot transformations (2D, 3D) involving scaling, rotation, and translation using homogeneous coordinates, along with sensor applications in robots
- Discuss robot cell design principles, including sequence control, operator interface, and safety monitoring devices. Briefly introduce mobile robot working principles and actuation using basic robotics software like MATLAB or NXT.

LECTURE WITH BREAKUP	**NO. OF LECTURES**
UNIT-I **INTRODUCTION:** Robot anatomy - Definition, law of robotics, History and Terminology of Robotics-Accuracy and repeatability of Robotics-Simple problems- Specifications of RobotSpeed of Robot-Robot joints and links-Robot classifications-Architecture of robotic systemsRobot Drive systems- Hydraulic, Pneumatic and Electric system.	5
UNIT-II **END EFFECTORS AND ROBOT CONTROLS:** Mechanical grippers-Slider crank mechanism, Screw type, Rotary actuators, cam type-Magnetic grippers-Vacuum grippers-Air operated grippers-Gripper force analysis-Gripper design-Simple problems-Robot controls-Point to point control, Continuous path control, Intelligent robot-Control system for robot joint-Control actions-Feedback devices-Encoder, Resolver, LVDT-Motion Interpolations-Adaptive control.	7
UNIT-III **ROBOT TRANSFORMATIONS AND SENSORS:** Robot kinematics-Types- 2D, 3D Transformation-Scaling, Rotation, Translation- Homogeneous coordinates, multiple transformation-Simple problems. Sensors in robots – Touch sensors-Tactile sensor – Proximity and range sensors – Robotic vision sensor-Force sensor-Light sensors, Pressure sensors.	6
UNIT-IV **ROBOT CELL DESIGN AND APPLICATIONS:** Robot work cell design and controlSequence Control, Operator interface, Safety monitoring devices in Robot-Mobile robot working principle, actuation using MATLAB, NXT Software Introductions-Robot applicationsMaterial handling, Machine loading and unloading, assembly, Inspection,Welding, Spray painting and undersea robot.	7
UNIT-V **MICRO/NANO ROBOTICS SYSTEM:** Micro/Nanorobotics system overview-Scaling effectTop down and bottom up approach- Actuators of Micro/Nano robotics system-Nanorobot communication techniques-Fabrication of micro/nano grippers-Wall climbing micro robot working principles-Biomimetic robot-Swarm robot-Nanorobot in targeted drug delivery system.	5
UNIT-VI **THE FUTURE OF ROBOTICS:** Current Research Trends in Robotics, Ethical Considerations Surrounding Autonomous Robots, The Potential Impact of Robotics on Society and Various Industries.	5

COURSE OUTCOMES
After completion of course, students would be able to: ● Students will be able to define robots and explain the Laws of Robotics. ● Students will be able to describe various end effectors used in robots and analyze their design considerations. ● Students will be able to apply basic transformations (scaling, rotation, translation) in the context of robot movement and understand the role of sensors in robot operations. ● Students will be able to explain the key elements of robot cell design and identify potential applications of robots in various industries ● Students will gain a foundational understanding of micro/nanorobotic systems, their actuation methods, and potential applications in healthcare or other fields.

References:

2. S.R. Deb, Robotics Technology and flexible automation, Tata McGraw-Hill Education., 2009
3. Mikell P Groover & Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, Industrial Robotics, Technology programming and Applications, McGraw Hill, 2012
4. Richard D. Klafter, Thomas .A, Chri Elewski, Michael Negin, Robotics Engineering an Integrated Approach, Phi Learning.,2009.
5. Francis N. Nagy, Andras Siegler, Engineering foundation of Robotics, Prentice Hall Inc., 1987.
6. P.A. Janaki Raman, Robotics and Image Processing an Introduction, Tata McGraw Hill Publishing company Ltd., 1995.
7. Carl D. Crane and Joseph Duffy, Kinematic Analysis of Robot manipulators, Cambridge University press, 2008.
8. Fu. K. S., Gonzalez. R. C. & Lee C.S.G., “Robotics control, sensing, vision and intelligence”, McGraw Hill Book co, 1987

MCE-P390 Minor Project /INTERNSHIP/ Industry Project

L T P Cr

0 0 6 8

Course Objectives: This course is designed to encourage design projects where students take what they have learned throughout the course of their ME program and apply it examine a specific idea.

Course Outcomes (CO): After the completion of this course the students will be able to:

1. Investigate and identity the real world problems
2. Design, develop and implement a domain specific design/research problem.
3. Develop acumen for higher education and research.
4. Enhance technical report writing skills.

Evaluation Scheme:

The student will be evaluated on the following points:
Literature Review
Identification of the Real world problem
Design, develop and implement the solution for the identified problem.
Discussion of Results and Inferences drawn
Conclusion
Presentation
Report Writing

MCE-S391 SEMINAR

L	T	P	Cr
0	0	4	4

Course Objectives: This course is designed to help the student obtain skills to discuss or present something within a group. Seminar Course is an outcome of six months of study, exploration, survey and analysis of a particular topic. It is designed to test the skills of the candidate in making a good presentation, Audience Engagement, Communication Skills. It also helps in building lifelong learning as a skill in the candidate.

Learning Outcomes (LOs)

LO1: Identification of a domain specific scholarly topic
LO2: Investigate and tabulate details and history about the selected topic
LO3: Application of the selected topic in domain or real life
LO4: Technical report writing
LO5: Demonstrating the communication skills by good presentation and engaging the audience.

Evaluation Scheme:

· Presenting a topic to an audience in a given time with a professionally prepared content.
· Literature Survey/Content: This includes the depth knowledge of the related work done by others related to Seminar Topic
· Viva (answer to the queries)
· Report Writing

MCE-P490 Dissertation with Research paper / INTERNSHIP/ Industry Project

L T P Cr

0 0 16 16

Course Objectives: This course is designed to help the student obtain research skills which includes a thorough survey of a particular domain, finding a research problem and presenting a methodology to resolve the problem; with adequate experimental results to strengthen the contribution. The students are also given an exposure where they learn to write research papers and presenting the work in the conferences. Students are also supposed to learn about communicating the impact of their work by different tools which includes video, poster and presentation.

Learning Outcomes (LOs)

LO1: Design and implementation of identified research problem or industrial projects.
LO2: Develop acumen for higher education and research.
LO3: Write technical reports and publish the research work in referred journals, national and international conferences of repute.
LO4: Foresee how their current and future work will influence/impact the economy, society and the environment.

Evaluation Scheme

The students are expected to complete the dissertation or industrial project. Evaluation of the students will be done on following points.

Literature Review
Identification of the research gaps/objectives
Methodology adopted
Discussion of Results and Inferences drawn
Conclusion and Future Scope.
Presentation Structuring
Usefulness/Contribution to the profession
Publication