ANNA UNIVERSITY, CHENNAI

NON- AUTONOMOUS COLLEGES

AFFILIATED TO ANNA UNIVERSITY

M.TECH., INFORMATION TECHNOLOGY

REGULATIONS 2025

PROGRAMME OUTCOMES (POs):

PO	Programme Outcomes
PO1	An ability to independently carry out research /investigation and development work to solve practical problems
PO2	An ability to write and present a substantial technical report/document.
PO3	Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PROGRAMME SPECIFIC OUTCOMES:

PSO1: Design, develop, and evaluate advanced IT solutions including software, network, and cloud-based systems using modern tools and technologies to address complex real-world problems.

PSO2: Conduct independent research in Information Technology, applying data analytics, artificial intelligence, and emerging IT methodologies to innovate, optimize, and contribute to the advancement of knowledge and industry practices.

ANNA UNIVERSITY, CHENNAI

POST GRADUATE CURRICULUM (NON.AUTONOMOUS AFFILIATED INSTITUTIONS)

Programme: M. Tech., Information Technology **Regulations:** 2025

Abbreviations:

BS – Basic Science (Mathematics)

ES – Engineering Science (Programme Core (**PC**), Programme Elective (**PE**))

SD – Skill Development

SL – Self Learning

OE – Open Elective

L – Laboratory Course

T – Theory

LIT – Laboratory Integrated Theory

PW – Project Work

TCP – Total Contact Period(s)

Semester I									
S. No.	**Course Code**	**Course Title**	**Type**	**Periods per week**			**TCP**	**Credits**	**Category**
				L	**T**	**P**			
1.	MA25C07	Advanced Mathematical Methods (CSIE)	T	3	1	0	4	4	BS
2.	CP25C01	Advanced Data Structures and Algorithms	LIT	3	0	4	7	5	ES (PC)
3.	CP25C02	Advanced Database Technologies	T	3	0	0	3	3	ES (PC)
4.	CP25C03	Advanced Operating Systems	T	3	0	0	3	3	ES (PC)
5.	NE25C01	Advanced Internet Technologies	T	3	0	0	3	3	ES (PC)
6.	IF25101	Technical Seminar	-	0	0	2	2	1	SD
						Total Credits	**22**	**19**	

Semester II									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	CP25C06	Cloud and Big Data Analytics	T	3	0	0	3	3	ES (PC)
2.	CP25C05	Artificial Intelligence and Machine Learning	LIT	3	0	0	3	3	ES (PC)
3.	IF25C01	Advanced Software Testing Techniques	LIT	3	0	2	5	4	ES (PC)
4.	CP25C07	Quantum Computing	T	2	0	0	2	2	ES (PC)
5.		Programme Elective I	T	3	0	0	3	3	ES (PE)
6.		Industry Oriented Course I	--	1	0	0	1	1	SD
7.	IF25201	Industrial Training	--	-	-	-	-	2	SD
8.		Self-Learning Course	--	-	-	-	-	1	-
						Total Credits	**17**	**19**	

Semester – III									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
THEORY									
1.		Programme Elective II	T	3	0	0	3	3	ES (PE)
2.		Programme Elective III	T	3	0	0	3	3	ES (PE)
3.		Programme Elective IV	T	3	0	0	3	3	ES (PE)
4.		Industry Oriented Course II	-	1	0	0	1	1	SD
5.		Open Elective	-	3	0	0	3	3	-
6.	IF25301	Project Work I	-	0	0	12	12	6	SD
						Total Credits	**25**	**19**	

Semester - IV									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	IF25401	Project Work II	---	0	0	24	24	12	SD
						Total Credits	**24**	**12**	

PROGRAMME ELECTIVE COURSES - (PE)

S. No.	Course Code	Course Title	Periods			TCP	Credits
			L	T	P		
1.	IF25001	Network Security and Risk Management	3	0	0	3	3
2.	IF25C02	MLOps	3	0	0	3	3
3.	IF25002	Computer Vision	3	0	0	3	3
4.	IF25003	Optimization Techniques	3	0	0	3	3
5.	IF25004	Cognitive Modelling	3	0	0	3	3
6.	IF25005	Natural Language Processing	3	0	0	3	3
7.	IF25006	Data Science and Bioinformatics	3	0	0	3	3
8.	IF25007	Deep Learning Techniques	3	0	0	3	3
9.	IF25008	Agile Methodology for Business Analysts	3	0	0	3	3
10.	IF25009	Advanced Graphics and Animation	3	0	0	3	3
11.	CP25C17	Edge and Fog Computing	3	0	0	3	3
12.	IF25C03	DevOps and Micro Services	3	0	0	3	3
13.	IF25010	Design Thinking and Patterns	3	0	0	3	3
14.	IF25011	Information and Cyber security	3	0	0	3	3
15.	IF25012	Geo Spatial Informatics	3	0	0	3	3
16.	IF25C04	Block Chain Technologies	3	0	0	3	3
17.	IF25013	Climate Change and Disaster Management	3	0	0	3	3
18.	IF25014	Financial Data Analytics	3	0	0	3	3
19.	IF25015	Ethical Hacking and Management	3	0	0	3	3
20.	IF25C05	Generative AI	3	0	0	3	3

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4

Course Objectives:

- Develop an in-depth understanding of advanced concepts in linear algebra, multivariate analysis, and number theory for computer science applications.
- Apply mathematical tools such as eigenvalue decomposition, SVD, and multivariate statistical methods to real-world computing and data-driven problems.
- Analyze and implement number-theoretic techniques for cryptography, security, and algorithmic problem-solving in computer science.

Linear Algebra: Vector spaces, norms, Inner Products, Eigenvalues using QR transformations, QR factorization, generalized eigenvectors, Canonical forms, singular value decomposition and applications, pseudo inverse, least square approximations.

Multivariate Analysis: Random vectors and matrices, Mean vectors and covariance matrices, Multivariate normal density and its properties, Principal components, Population principal components, Principal components from standardized variables.

Elementary Number Theory: The division algorithm, Divisibility and the Euclidean algorithm, The fundamental theorem of arithmetic, Modular arithmetic and basic properties of congruences; Principles of mathematical induction and well ordering principle. Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence.

Advanced Number Theory: Advanced Number Theory, Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithm, Factorization Methods, Side Channel Attacks, Shannon Theory, Perfect Secrecy, Semantic Security.

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%.

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Gilbert Strang, Linear Algebra and Its Applications, Cengage Learning.
2. Richard A. Johnson & Dean W. Wichern, *Applied Multivariate Statistical Analysis*, Pearson.
3. Neal Koblitz, A Course in Number Theory and Cryptography*,* Springer.
4. Victor Shoup, A Computational Introduction to Number Theory and Algebra, Cambridge University Press.

E-resources:

1. https://ocw.mit.edu/courses/18-06-linear-algebra
2. https://nptel.ac.in/courses/111105041
3. https://crypto.stanford.edu/pbc/notes/numbertheory

CP25C01	Advanced Data Structures and Algorithms	L	T	P	C
		3	0	4	5

Course Objectives:

- To explore advanced linear, tree, and graph data structures and their applications.
- To design efficient algorithms using appropriate algorithmic paradigms.
- To evaluate computational complexity and identify tractable vs. intractable problems.

Linear Data Structures and Memory Optimization: Advanced arrays: Sparse arrays, dynamic arrays, cache-aware structures, Linked lists: Skip lists, unrolled linked lists, XOR linked lists, Stacks and Queues: Priority queues, double-ended queues, circular buffers, Hashing: Perfect hashing, cuckoo hashing, extendible hashing.

Practical:

- Implement skip lists and measure performance compared with balanced BST.
- Experiment with cache-aware data structures and analyze memory utilization.

Advanced Tree Data Structures: Balanced Trees: AVL, Red-Black Trees, Splay Trees, Treaps, Multi-way Trees: B-Trees, B+ Trees, R-Trees, Segment Trees, Fenwick Trees, Suffix Trees and Tries for string processing, Applications in indexing, text retrieval, computational geometry.

Practical:

- Implement B+ tree for database indexing use-case.
- Design a suffix tree-based algorithm for DNA sequence matching.

Graph Data Structures and Algorithms: Representation: Adjacency list/matrix, incidence matrix, compressed storage, Traversals: DFS, BFS with applications, Shortest Path Algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson's algorithm, Minimum Spanning Trees: Prim's, Kruskal's, Borůvka's algorithm, Network Flow Algorithms: Ford-Fulkerson, Edmonds-Karp, Push-Relabel.

Practical:

- Implement Johnson's algorithm for sparse graph shortest paths.
- Demonstration of Maximum flow in traffic or network routing simulation.

Algorithm Design and Paradigms: Divide and Conquer: Karatsuba's multiplication, Strassen's algorithm, Greedy Methods: Huffman coding, interval scheduling, set cover approximation, Dynamic Programming: Matrix chain multiplication, Floyd-Warshall, knapsack variants, Backtracking and Branch-and-Bound, Randomized Algorithms and Probabilistic Analysis.

Practical:

- Implement Strassen's algorithm and compare with naive matrix multiplication.
- Develop a randomized algorithm for primality testing (Miller–Rabin).

Computational Complexity and Approximation Algorithms: Complexity Classes: P, NP, NP-Complete, NP-Hard, Reductions: Polynomial-time reductions, Cook-Levin theorem (overview), Approximation Algorithms: Vertex cover, set cover, TSP, k-center problem, Heuristic Algorithms: Local search, simulated annealing, genetic algorithms.

Practical:

- Implement approximation algorithm for vertex cover.
- Complexity analysis of a chosen NP-hard problem and implement a heuristic.

Advanced Topics and Emerging Trends: Randomized Algorithms – Monte Carlo Algorithms, Parallel and Distributed Algorithms – PRAM Model, Divide and Conquer in Parallel, Load Balancing, Streaming Algorithms – Data Stream Models, Sketching and Sampling, Frequency Moments, Advanced String Matching – Suffix Trees, Suffix Arrays, Pattern Matching in Linear Time.

Practical:

- Implement randomized and streaming algorithms on real-world datasets.
- Design of parallel and distributed algorithms.

Weightage: Continuous Assessment: 50%, End Semester Examinations: 50%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

References:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT Press.
2. La Rocca, M. (2021). Advanced algorithms and data structures. Manning Publications.
3. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). Data structures and algorithms in C++. John Wiley & Sons, Inc.
4. Weiss, M. A. (2014). Data structures and algorithm analysis in C++. Pearson Education.
5. Drozdek, A. (2013). Data structures and algorithms in C++. Cengage Publications.

E-resources:

1. https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/
2. https://github.com/afrid18/Data_structures_and_algorithms_in_cpp
3. https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-c/?srsltid=AfmBOorEZlkgV7QzaEh6IqzAaKLjC-IpFU1NGgWFoHMLhOos-uDVKjCK

	Description of CO	**PO**	**PSO**
CO1	Describe data structures and implement algorithmic solutions for complex computational problems.	--	--
CO2	Analyze the time complexity and efficiency of algorithms for various computing problems.	PO1(3)	PSO1(3)
CO3	Evaluate algorithmic techniques and data structures to determine their suitability for different applications.	PO3(2)	PSO2(2)
CO4	Design optimized solutions for real-world problems using appropriate algorithms and data structures.	PO2(1)	PSO1(3)

CP25C02	**Advanced Database Technologies**	L	T	P	C
		3	0	0	3

Course Objectives:

- To strengthen the understanding of enhanced ER models and their transformation into relational models with indexing and file structures.
- To understand object-oriented and object-relational database concepts and querying using OQL.
- To explore techniques in query processing, execution, and optimization strategies.

Entity Relationship Model: Entity Relationship Model Revised-Subclasses, Superclasses and Inheritance -Specialization and Generalization-Union Types-Aggregation.

Activity: Design ER Model for a specific use case.

Enhanced Entity Relational Model: Relational Model Revised, Converting ER and EER Model to Relational Model-SQL and Advanced Features, File Structures, Hashing, and Indexing.

Activity: Demonstration of SQL Implementation.

Object Relational Databases: Object Database Concepts-Object Database Extensions to SQL, The ODMG Object Model and ODL, Object Database Conceptual Design-Object Query Language OQL-Language Binding in the ODMG Standard.

Activity: Demonstration of Object Query Language.

Query Processing and Optimization: Query Processing, Query Trees and Heuristics, Query Execution Plans, Cost Based Optimization.

Activity: Design of Query Evaluation Plans.

Distributed Databases: Real-Time Bidding, E-mail Marketing, Affiliate Marketing, Social Marketing Mobile Marketing, Distributed Database Concepts, Data Fragmentation, Replication and Allocation, Distributed Database Design Techniques, Distributed Database Design Techniques, Distributed Database Architectures.

Activity: Demonstration of Concurrency and Transactions.

NOSQL Systems and Bigdata: Introduction to NOSQL Systems-The CAP Theorem, Document, based NOSQL Systems, Key-value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j.

Activity: Design application with MongoDB.

Advanced Database Models, Systems and Applications: Active Database Concepts and Triggers, Temporal Database Concepts, Spatial Database Concepts, Multimedia Database Concepts, Deductive Database Concepts, Introduction to Information Retrieval and Web Search.

Activity: Demonstration of Hadoop infrastructure for Big Data Analytics.

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems. Pearson Education.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database system concepts, McGraw Hill Education.
3. Ceri, S., & Pelagatti, G. Distributed databases: Principles and systems. McGraw Hill.
4. Ramakrishnan, R., & Gehrke, J. (2004). Database management systems. McGraw Hill.

E-resources:

1. https://www.edx.org/learn/sql/stanford-university-databases-advanced-topics-in-sql
2. https://www.coursera.org/courses?query=sql&productDifficultyLevel=Advanced

	Description of CO	PO	PSO
CO1	Elaborate different database models for effective database design.	--	--
CO2	Implement advanced database features for optimized data retrieval.	PO1(3)	PSO1(3)
CO3	Evaluate query processing and optimization strategies to improve system performance.	PO3(2)	PSO2(2)
CO4	Design solutions using advanced database models to address complex data-intensive applications.	PO2(1)	PSO1(3)

CP25C03	**Advanced Operating Systems**	L	T	P	C
		3	0	0	3

Course Objectives:

- To analyze the architectures and design issues of advanced operating systems.
- To develop the model for process synchronization and recovery in complex environments.
- To evaluate algorithms for distributed coordination, resource management, fault tolerance, and security.

Advanced Process and Thread Management: Multithreading models, thread pools, context switching, Synchronization issues and solutions: semaphores, monitors, lock-free data structures, CPU scheduling in multi-core systems

Activity: CPU scheduler simulation for multicore systems.

Memory and Resource Management in Modern OS: Virtual memory, demand paging, page replacement policies-Huge pages, NUMA-aware memory management-Resource allocation in cloud-native environments

Activity: Simulate demand paging and page replacement algorithms.

Virtualization and Containerization: Hypervisors (Type I & II), KVM, QEMU, Xen-Containers: Docker, LXC, systemd-nspawn-OS-level virtualization and namespaces

Activity: Deploy and configure Docker containers with various images.

Distributed Operating Systems and File Systems: Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance

Activity: Simulate distributed process synchronization.

Security and Trust in Operating Systems: Access control models: DAC, MAC, RBAC-OS hardening techniques, sandboxing, SELinux, AppArmor-Secure boot, rootkit detection, trusted execution environments

Activity: Implement Role-Based Access Control (RBAC) using Linux user and group permissions.

Real-Time and Embedded Operating Systems: Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies

Activity: Analyze FreeRTOS task scheduling behavior.

Edge and Cloud OS: Future Paradigms: Serverless OS, unikernels, lightweight OS for edge computing-Mobile OS internals (Android, iOS)-OS for quantum and neuromorphic computing (intro)

Activity: Analyze Android's system architecture using emulator tools.

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Tanenbaum, A. S., & Bos, H. (2023). Modern operating systems. Pearson.
2. Buyya, R., et al. (2022). Content delivery networks and emerging operating systems. Springer.
3. Silberschatz, A., Galvin, P. B., & Gagne, G. (2022). Operating system concepts. Wiley.
4. Anderson, T., & Dahlin, M. (2021). Operating systems: Principles and practice. Recursive Books.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2020). Operating systems: Three easy pieces.

E-Resources:

1. Prof. Smruti Ranjan Sarangi, "Advanced Distributed Systems", IIT Delhi, NPTEL, https://onlinecourses.nptel.ac.in/noc22_cs80/preview
2. Prof. Rajiv Misra, "Cloud Computing and Distributed Systems", IIT Patna, NPTEL, https://nptel.ac.in/courses/106104182

	Description of CO	**PO**	**PSO**
CO1	Describe operating system concepts for memory and resource management.	--	--
CO2	Analyse virtualization and distributed OS mechanisms for scalability and performance.	PO1(3)	PSO1(3)
CO3	Evaluate OS security and resource handling strategies in diverse environments.	PO3(2)	PSO2(2)
CO4	Design innovative OS solutions using modern tools and techniques.	PO2(1)	PSO1(3)

		L	T	P	C
NE25C01	**Advanced Internet Technologies**	3	0	0	3

Course Objective:

- To provide knowledge about modern Internet technologies and applications.
- To enable students to design, implement, and evaluate web-based and distributed applications.
- To develop skills in applying security mechanisms and scalable architectures for real-world Internet-based solutions.

Internet and Web Design: Understanding HTML and XHTML Connections, Understanding Cascading Stylesheets, Understanding JavaScript, Working with Fonts, Texts and Lists, Tables, Internal and External Links, Working with colours and Multimedia.

Activity: Design a responsive webpage using HTML5 and CSS3.

CSS and Dynamic websites: Understanding CSS Box Model and positioning, Navigations, creating fixed and liquid layouts, understanding Dynamic websites.

Activity: Demonstrate RESTful APIs using HTTP methods

Java Script Programming: Working with DOM, Java Script variables, strings and arrays, Functions and objects, Controls and loops, Responding to events, windows and frames using Un obtrusive Java script, using third party libararies, AJAX.

Activity: Build a dynamic SPA using JavaScript.

Emerging Internet Technologies – I: Cookies and sessions, Middleware, Sending Email, production Concerns, Persistence, Routing, Rets APIs and JSON.

Activity: Demonstration using REST APIs and JSON.

Emerging Internet Technologies-II: Static Content, Implementing MVC in express, Security Integrating third party APIs, Debugging, Going live, Maintenance.

Activity: Design of simple login system with JWT authentication

Flask web development in Internet Technologies: Basic Application structure, Templates, Databases, Email, Large Application structure, Application Programming Interfaces, Testing, Deployment.

Activity: Build a simple web application using Flask APIs.

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Brown, E. (2023). Web development with Node and Express: Leveraging the JavaScript stack. O’Reilly Media.
2. Wargo, J. M. (2021). Learning progressive web apps: Building modern web apps using service workers and web APIs. Addison-Wesley Professional.
3. Meloni, J. C. (2023). HTML, CSS, and JavaScript all in one . Sams Publishing.
4. Grinberg, M. (2018). Flask web development: Developing web applications with Python O’Reilly Media.
5. Vemula, R. (2023). Real-time web application development. Packt Publishing.

	Description of CO	**PO**	**PSO**
CO1	Explain web technologies to create responsive and interactive web pages.	--	--
CO2	Develop dynamic web applications and third-party libraries for enhanced user experience.	PO1(3)	PSO1(3)
CO3	Integrate middleware and IoT components to build scalable and secure internet-based applications.	PO3(2)	PSO2(2)
CO4	Design and deploy full-stack applications using Flask.	PO2(1)	PSO1(3)

CP25C06	**Cloud and Big Data Analytics**	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand cloud computing paradigms and service models.
- To study big data characteristics, technologies and storage.
- To explore Hadoop ecosystem and processing tools.
- To analyze, store, and visualize large-scale data.
- To develop and deploy cloud-based big data applications.

Cloud Computing: Introduction to Cloud Computing - Service Models: IaaS, PaaS, SaaS - Deployment Models: Public, Private, Hybrid, Community - Virtualization: Hypervisors, VM provisioning - Cloud providers: AWS, Azure, Google Cloud overview

Activities: Create a free-tier AWS account and explore EC2, S3.

Big Data Ecosystem & Hadoop: Big Data characteristics (Volume, Velocity, Variety, Veracity, Value) - Challenges in Big Data - Hadoop Architecture and HDFS - MapReduce Programming Model –YARN

Activities: Watch Hadoop tutorial and submit a summary report.

Data Storage & Processing Techniques: NoSQL Databases: MongoDB, Cassandra, HBase - Data Ingestion: Sqoop, Flume - Hive, Pig – Architecture and Querying - Data Cleaning and Preparation.

Activities: HiveQL assignments on datasets from Kaggle.

Advanced Big Data Analytics: Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs- Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming

Activities: Mini-project using PySpark on COVID or Traffic dataset.

Cloud Integration and Security: Data on Cloud – Storage and Compute Services - Data Security and Privacy in Cloud - Cloud-based Data Analytics Use Cases - Cloud-Native vs Traditional Applications - Case Study: Smart Cities / Healthcare Analytics

Activities: Create a cloud deployment report using Azure/AWS.

Weightage: Continuous Assessment:40%, End Semester Theory Examination: 60%

Assessment Methodology: Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley.
2. Tom White, Hadoop: The Definitive Guide, O'Reilly Media.
3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing.
4. Kleppmann, Martin, Designing Data-Intensive Applications, O'Reilly.
5. Alan Gates, Programming Pig, O'Reilly.
6. Jure Leskovec et al., Mining of Massive Datasets, Cambridge University Press.

E-resources:

https://www.nasa.gov/smallsat-institute/sst-soa/thermal-control/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and service models of cloud computing and big data analytics used in modern data-driven environments.	--	--
CO2	**Analyze** cloud-based big data processing frameworks and analytics techniques to understand their performance, scalability, and resource utilization.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** big data analytics tools, cloud platforms, and data processing models for their effectiveness in handling large-scale and complex datasets.	PO3 (2)	PSO2 (2)
CO4	**Design** scalable cloud-based big data analytics solutions by selecting appropriate architectures, platforms, and analytical methods for real-world applications.	PO2 (1)	PSO1 (3)

CP25C05	**Artificial Intelligence and Machine Learning**	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamentals of Artificial Intelligence and its core problem-solving techniques using intelligent agents, search strategies, and logic-based reasoning.
- To comprehend the theoretical foundations and algorithmic frameworks of machine learning, with an emphasis on supervised and unsupervised learning methodologies.

Problem Solving and Knowledge Representation: Solving problems by searching – adversarial search – constraint satisfaction problems – Logical agents – Propositional logic – First order logic – Forward chaining – Backward chaining – Ontological Representations and Reasoning Systems.

Activities: Develop a simple Intelligent Agent that navigates a grid world using search algorithms.

Uncertain Knowledge and Reasoning: Overview of uncertainty and basic probability – Baye's rule – Bayesian networks – Hidden Markov models – Kalman filters – Utility Functions – Decision Networks – Sequential decision problems – Game theory

Activities: Develop a simple Intelligent Agent that makes decisions in adversarial conditions

Machine Learning Introduction: Types of learning – Hypothesis space – Inductive bias – Evaluation – Cross validation – Bias variance – Bias Variance Trade off.

Activities: Develop a simple Intelligent Agent that uses logical reasoning to infer hidden facts or constraints

Supervised Learning: Linear Regression – Logistic Regression - Decision trees: Classification and regression trees – Neural networks, multilayer perceptron – Support vector machines: linear and non-linear kernel functions – K nearest neighbours – Ensemble learning: bagging and boosting.

Activities: Behaviour Modelling, Disease Diagnosis and any other similar ones suitable for supervised learning

Unsupervised and Probabilistic Learning: Clustering: partition, hierarchical and density based clustering – Self organizing maps – Expectation maximization – Gaussian mixture models – Principal component analysis – Bayesian learning: Bayes optimal classifier, Naïve Bayes classifier, Bayesian belief networks.

Activities: Customer Segmentation, Image Processing and any other similar ones suitable for unsupervised and probabilistic learning

Weightage: Continuous Assessment: 40%, End Semester Theory Examination: 60%

Assessment Methodology: Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Stuart Russell and Peter Norvig Artificial Intelligence - A Modern Approach, Prentice Hall, 3rd edition, 2011.
2. Elaine Rich, Kevin Knight and Shiv Shankar B. Nair, Artificial Intelligence, 3rd edition, Tata McGraw Hill, 2009.
3. Tom Mitchell, “Machine Learning”, McGraw Hill, 3rd Edition, 1997.
4. Ethem Alpaydin,"Introduction to Machine Learning”, MIT Press, Prentice Hall of India, Third Edition 2014.
5. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar "Foundations of Machine Learning”, MIT Press, 2012.

E-resources:

1. Nptel Course, An Introduction to Artificial Intelligence by Prof Mausam
2. Nptel Course, Machine Learning,By Prof. Carl Gustaf Jansson

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, scope, and applications of artificial intelligence and machine learning in solving real-world problems.	--	--
CO2	**Analyze** various artificial intelligence and machine learning models to understand their behavior, performance, and suitability for different problem domains.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** machine learning algorithms and artificial intelligence techniques based on accuracy, efficiency, and applicability to specific use cases.	PO3 (2)	PSO2 (2)
CO4	**Design** intelligent systems by selecting and integrating appropriate machine learning models and AI techniques to address complex real-world applications.	PO2 (1)	PSO1 (3)

IF25C01	Advanced Software Testing Techniques	L	T	P	C
		3	0	2	4

Course Objective:

To provide students with a deep understanding of modern and advanced software testing techniques, including exploratory testing, test heuristics, risk analysis, and real-world defect detection strategies. This course emphasizes hands-on problem-solving and designing robust testing strategies.

Foundations and Testing Thinking

Role of the Software Tester-Common Testing Fallacies-Challenges in Software Testing-Characteristics of Good Tests-Developing a Tester's Mindset-Heuristics in Software Testing-Risk Based Testing Approaches-Importance of Negative Testing-Software Modeling for Testing

Activities: Understand the Role of a Tester and Identify Risks, Design Good Tests using Test Heuristics

Advanced Test Design Techniques

Exploratory and Scripted Testing-Equivalence Partitioning and Boundary Value Analysis-State Transition Testing Techniques-Decision Tables and Pairwise Testing-Stress and Load Testing Methods-Approaches to Security Testing-Regression Test Planning-Designing Test Automation-Effective Bug Reporting Practices

Activities: Perform Boundary Value, Equivalence, and Pairwise Testing

Exploratory Testing in Practice

Overview of Exploratory Testing-Charter-Based Testing Sessions-Test Tour Techniques-Investigative Testing with Models-Observation and Note-Taking Techniques-Use of Oracles and Heuristics-Session-Based Test Management (SBTM)-Debriefing and Reporting in Exploratory Testing-Designing Focused Test Missions

Activities: Conduct Exploratory Testing with Session Notes, Practice Bug Reporting and Use Oracles, Perform Load and Stress Testing

Automation and Strategy Integration

Guidelines for Test Automation-Automation in Exploratory Testing-Test Framework Selection and Design-Continuous Testing in DevOps Environments-Testing in Agile and CI/CD-Workflows-Behavior-Driven Development (BDD) Techniques-Analyzing Test Coverage and Identifying Gaps-Test Metrics and Dashboarding Tools-Managing Technical Debt in Testing

Activities: Design and Automate Regression Tests, Use BDD for Agile Testing

Domain-Specific and Specialized Testing

Mobile Application Testing-Testing Web and Cloud-Based Applications-Usability and Accessibility Testing-Testing for Embedded and IoT Systems-Game Testing

Methodologies-Testing AI/ML-Based Software-Ethical Considerations in Software Testing-Conducting Test Strategy Reviews and Audits-Emerging Trends in Software Testing

Activities: Perform Mobile and Web App Testing, Review Test Strategy and Metrics Dashboard

Course Outcomes:

1. Analyze and apply various advanced testing strategies to real-world problems.
2. Design and execute effective exploratory testing sessions.
3. Apply test heuristics and risk-based strategies in complex systems.
4. Integrate testing techniques into CI/CD and Agile environments.
5. Automate testing using appropriate tools and frameworks effectively.

Weightage: Continuous Assessment: 60%, End Semester Theory Examination: 40%

(i) Activity: 15% (ii) Internal Theory Examination: 35% (iii) Internal Laboratory Examinations: 15%
Assessment Methodology: Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Cem Kaner, James Bach, Bret Pettichord, Lessons Learned in Software Testing, Wiley, 2002.
2. Elisabeth Hendrickson, Explore It! Reduce Risk and Increase Confidence with Exploratory Testing, Pragmatic Bookshelf, 2013.
3. Rex Black, Advanced Software Testing – Vol 2, Rocky Nook, 2009.
4. Alan Richardson, Dear Evil Tester, Self-published, 2016.
5. Lisa Crispin & Janet Gregory, Agile Testing Condensed, Agile Testing Fellowship, 2019.

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, methodologies, and advanced practices of software testing used to ensure high-quality software systems.	--	--
CO2	**Analyze** advanced software testing techniques and test strategies to identify defects, assess risks, and improve software reliability.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** software testing tools, automation frameworks, and testing models to determine their effectiveness in complex software environments.	PO3 (2)	PSO2 (2)
CO4	**Design** comprehensive and efficient software testing solutions by selecting appropriate techniques, tools, and automation strategies for real-world applications.	PO2 (1)	PSO1 (3)

CP25C07	**Quantum Computing**	L	T	P	C
		2	0	0	2

Course Objective:

1. To provide a mathematical foundation for Quantum Computing and provide the basics of working
2. To interpret the various aspects and applications of quantum computing.
3. To examine the factors that affect Quantum computation

Physical properties and mathematical foundations

Double Slit Experiment; Light: Particle Vs Wave; Heisenberg Uncertainty Principle.Vector spaces – basis; Inner product; Outer product; Tensor product; Linear operators

Activities:

- Simulate the experiment using an interactive virtual lab
- Construct simple operators and visualize action on vectors

Quantum computing postulates and gates

Review of postulates, Bloch sphere, Single qubit states and gates, superposition; Two Qubit States and Gates - Bell States, Entanglement, CNOT gate, Phase oracles, Pauli Gates.

Activities:

- Group quiz to match postulates to physical implications
- Visualization with Bloch sphere simulators

Quantum computing circuits

Dirac’s notation for quantum computing, Computational Basis, Orthonormality, Hadamard and Phase Gates- building quantum circuits

Activities:

- Use IBM Q Composer to build and simulate custom circuits
- Hands-on: Apply X, H, Z gates and observe results on simulators

Fundamental Quantum Algorithms

Deutsch–Josza Algorithm, Grover search algorithm: Problem definition, Amplitude amplification, Grover oracle, diffuser, multiple solutions in the search space

Activities:

- Construct DJ circuit for a 3-bit input function
- Simulation of Grover’s algorithm with multiple marked elements

Programming on a real quantum computer

Coding a real time quantum computer via IBMQ to carry out basic quantum measurement and state analysis.

Activities:

- Connect Qiskit with IBMQ using personal API token
- Hands-on: Create 1- and 2-qubit circuits using Hadamard, X, Z, and measurement gates
- Compare real and simulated results and Observe impact of quantum noise

Text Books:

1. Chuck Easttom, “Quantum Computing Fundamentals”, 1st edition, Published by Addison-Wesley Professional (June 1st 2021)
2. Qiskit TextBook - https://qiskit.org/textbook/preface.html (2022)

References:

1. Kasirajan, Venkateswaran. Fundamentals of quantum computing. Springer International Publishing, 2021.
2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020
3. Nielsen, Michael A., and Isaac L. Chuang, “Quantum Computation and Quantum Information” Cambridge University Press (5 April 2013)

Evaluation method to be used

Continuous Assessment	Activities	End Semester
50%	10%	40%

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, postulates, and computational models of quantum computing and their significance in next-generation computing systems.	--	--
CO2	Analyze quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	Evaluate quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.	PO3 (2)	PSO2 (2)
CO4	Design quantum circuits and algorithmic solutions by selecting appropriate quantum gates, qubit architectures, and computational models for real-world problem scenarios.	PO2 (1)	PSO1 (3)

IF25001	**Network Security and Risk Management**	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the principles and protocols behind network security.
- Identify, assess, and mitigate security threats and risks in enterprise networks.
- Apply risk management frameworks to real-world security policies.

Foundations of Network Security: Security Goals - CIA - Threats, Vulnerabilities, and Attack Vectors - OSI and TCP/IP Security Issues - Firewalls, IDS/IPS, and NAT Basics - Introduction to Zero Trust Architecture. Wireshark traffic analysis - Basic firewall rule design.

Activities:

- Case study-based group activity to identify threats, vulnerabilities, and mitigations in a sample system.
- Small-group activity to design a Zero Trust model for a fictional organization (tools: diagrams, mock configs).
- Map typical threats and mitigations to OSI and TCP/IP layers.

Cryptography for Secure Communication: Symmetric vs. Asymmetric Cryptography - DES, AES, RSA, ECC, Diffie-Hellman - Digital Signatures and Certificates - TLS/SSL Protocol Internals - Public Key Infrastructure (PKI) & Certificate Management. OpenSSL for hands-on encryption/decryption - Certificate generation using OpenCA or XCA.

Activities:

- Symmetric vs. Asymmetric Encryption Demo (OpenSSL)
- Encrypt and decrypt files using AES (symmetric) and RSA (asymmetric).
- Certificate Generation with OpenCA or XCA
- Create a root CA, generate and sign certificates, and validate them.

Enterprise Network Security Architecture: Enterprise Segmentation and DMZ Design - Secure Routing Protocols (RIP, OSPF, BGP Security - VPNs (IPSec, SSL VPN) - Wireless Network Security (WPA2/3, EAP) - Authentication: RADIUS, TACACS+. VPN setup using WireGuard/OpenVPN - WPA2 Enterprise configuration using FreeRADIUS

Activities:

- Secure Routing Protocols Simulation
- Simulate OSPF or RIP with authentication in GNS3 or Packet Tracer.
- WPA2-Enterprise Setup with Free RADIUS
- Configure a small test network using FreeRADIUS and WPA2-Enterprise authentication

Risk Management and Security Policy: Risk Assessment Methodologies (OCTAVE, NIST RMF) - Threat Modeling (STRIDE, DREAD) - Business Continuity Planning (BCP) and Disaster Recovery (DR) - Security Policies, Standards, and Guidelines (ISO/IEC 27001, NIST 800 series) - Legal & Ethical Aspects: GDPR, HIPAA.

Activities:

- Risk Assessment Workshop (OCTAVE or NIST RMF)
- Use a case study to perform qualitative risk analysis and identify key controls.
- Threat Modelling Exercise (STRIDE/DREAD)
- Apply STRIDE to a sample application or system to uncover potential threats.

Advanced and Emerging Trends in Security: Security in Cloud and Edge Computing - Software-Defined Networking (SDN) Security - IoT and 5G Network Security Challenges - Blockchain in Network Security - Cyber Threat Intelligence (CTI) & SIEM.

Activities:

- IoT Risk Assessment Activity
- Evaluate the security of a mock IoT system and identify mitigation strategies.
- Blockchain Use-Case Simulation
- Develop a simple proof-of-concept for secure DNS or access logging using a blockchain framework.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. W. Stallings, Network Security Essentials: Applications and Standards, 6th ed. Pearson, 2017.
2. MIT OpenCourseWare, Computer Systems Security (6.858), Massachusetts Institute of Technology.
3. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York, NY, USA: Wiley, 1996.
4. Cisco Systems, Cisco SAFE: A Security Blueprint for Enterprise Networks, White Paper, Cisco Systems, Inc., San Jose, CA, USA, 2015.
5. European Union Agency for Cybersecurity (ENISA), Risk Management: Implementation Principles and Inventories for Risk Management/Risk Assessment Methods and Tools, Nov. 2006.

E-resources:

1. NPTEL - Network Security by Dr. D. Mukhopadhyay (IIT Kharagpur)
2. NPTEL - Cryptography and Network Security by Prof. D. Mukhopadhyay

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and components of network security and risk management in modern information systems.	--	--
CO2	**Analyze** network security threats, vulnerabilities, and risk assessment techniques to protect organizational information assets.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** network security mechanisms, policies, and risk mitigation strategies to determine their effectiveness in securing network infrastructures.	PO3 (2)	PSO2 (2)
CO4	**Design** secure network architectures and risk management frameworks by selecting appropriate security controls and mitigation strategies for real-world environments.	PO2 (1)	PSO1 (3)

IF25C02	**MLOps**	L	T	P	C
		3	0	0	3

Course Objectives:

- This course equips students with the necessary tools and expertise to become proficient MLOps engineers, addressing the industry's shortage of skilled professionals capable of deploying and managing machine learning models efficiently.
- Train & Deploy ML models, Monitor, Retrain & Improve, and Scale AI applications.

MLops Fundamentals: Introduction to MLOps-purpose and benefits of MLOps, role in accelerating the ML development process, impact on model deployment and maintenance, MLOps Principles-continuous integration, continuous deployment, version control, and automation, ML Lifecycle- various stages of ML lifecycle, data collection, preprocessing, model training, validation, deployment, and monitoring, MLOps Architecture.

Activities:

- **Group Discussion/Presentation:** On benefits of MLOps vs traditional ML workflows.
- **Case Study Analysis:** Analyze a real-world MLOps implementation (e.g., Netflix, Google, or Uber).

- **Roleplay Exercise:** Assign roles (Data Engineer, ML Engineer, DevOps) and simulate an ML project kickoff meeting.
- **Mini Quiz:** Concepts of CI/CD, version control, automation in ML

Data Management and Preparation: Data Collection and Validation-gather, clean, and validate data for ML models, Feature Engineering and Selection-extracting meaningful features from data and selecting the most important ones for model training, Data Versioning-importance of managing data versions for reproducibility and auditing, Data Pipelines-Building robust data pipelines for automating data processing and feeding data to models.

Activities:

- **Data Cleaning Exercise:** Provide a raw dataset for students to clean and validate.
- **Feature Engineering Challenge:** Students extract and select relevant features from a dataset.
- **Data Versioning Task:** Use tools like DVC to version a dataset and demonstrate reproducibility.

Model Building and Experimentation: Model Selection and Training-Choosing the right ML algorithm and training models on the prepared data, Experiment Tracking-using tools like MLflow to track ML experiments and their results, Model Resource Management-Optimizing model resource usage and deployment strategies, Model Evaluation and Tuning-Evaluating model performance and tuning hyper parameters for optimal results.

Activities:

- **Model Comparison Exercise:** Train 2–3 different ML models on the same dataset and compare results.
- **Leaderboard Competition:** Mini Kaggle-style competition where students track and tune their models.
- **Model Evaluation Drill:** Perform confusion matrix, precision, recall, ROC-AUC, etc., on a classification model.

Deployment and Monitoring: Model Serving-models serving patterns and infrastructure, Containerization with Docker- Using Docker to package and deploy ML models, CI/CD Pipelines: Automating ML model training and deployment using CI/CD pipelines, Model Monitoring: Implementing monitoring systems to track model performance and identify potential issues.

Cloud Deployment: Deploying ML models on cloud platforms like Google Cloud, Amazon AWS, or Azure.

Activities:

- **Cloud Lab:** Deploy a model to Google Cloud, AWS, or Azure (free tier).

- **Monitoring Simulation:** Set up basic logging and alerting for a deployed model.
- **Scaling Exercise:** Simulate traffic to a deployed model and scale it using Kubernetes (or simulated with Docker Compose).

Tools and Technologies: MLflow-MLflow for experiment tracking, model management, and deployment, TensorFlow/PyTorch-popular ML frameworks, Kubernetes-Kubernetes for deploying and managing ML applications, Cloud Platform Tools-tools offered by cloud providers for MLOps, Databricks- Exploring Databricks for data science and MLOps, Vertex AI: Using Vertex AI for MLOps on Google Cloud, Practical application and projects

Activities:

- **Tool Comparison Matrix:** Compare features and use cases of MLflow, Vertex AI, Databricks, TensorFlow, PyTorch, and Kubernetes.
- **Framework Implementation** Task: Build and train a model using both TensorFlow and PyTorch.
- **End-to-End Mini Project:** Students build, train, deploy, and monitor a small model using the full MLOps stack.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Yaron Haviv, Noah Gift, Implementing MLOps in the Enterprise 2023
2. Emmanuel Raj, Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale 2021.
3. Noah Gift, Alfredo Deza, Practical MLOps 2021.
4. Stephen Fleming, Accelerated DevOps with AI, ML & RPA: Non-Programmer's Guide to AIOPS & MLOPS 2020.
5. Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann, Introducing MLOps, 2020.

E-resources:

1. MLOps Principles –MLflow,Bloghttps://mlflow.org/docs/latest/index.html
2. Machine Learning and Deep Learning Fundamentals and Applications (IIT Guwahati – Prof. M. K. Bhuyan)

CO	Description of CO	PO	PSO

CO1	Describe the fundamental concepts, workflows, and lifecycle management practices of MLOps for deploying and maintaining machine learning systems.	--	--
CO2	**Analyze** MLOps pipelines and automation practices to understand model deployment, monitoring, versioning, and performance management.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** MLOps tools, platforms, and operational strategies to assess their effectiveness in managing scalable and reliable machine learning solutions.	PO3 (2)	PSO2 (2)
CO4	**Design** end-to-end MLOps solutions by selecting appropriate tools, architectures, and best practices for real-world machine learning deployments.	PO2 (1)	PSO1 (3)

IF25002	**Computer Vision**	L	T	P	C
		3	0	0	3

Course Objective:

1. Learn fundamentals of and techniques used in image processing and computer vision.
2. Understand Segment the image to identify the region of interest
3. Identify various algorithms for Motion Estimation & Pattern recognition.
4. Develop an algorithm to recognize the specified objects in the given image.
5. To recognize and describe both the theoretical and practical aspects of computing with images.
6. To learn pattern recognition.

Introduction: Image Formation Models, Orthographic & Perspective Projection, Geometric Transformations and Image Warping, Image Representation Schemes, Camera Models: Pinhole Camera, Perspective Projection and Camera calibration, Image Sampling and Quantization.

Activities:

- **Mini Project:** Calibrate a simple pinhole camera model using OpenCV. Estimate focal length, principal point, and distortion coefficients using checkerboard calibration.
- **Quiz/Worksheet:** Identify projection types (orthographic vs. perspective) using real-world images.
- **Group Activity:** Compare and present image representation formats (e.g., RAW, BMP, JPEG) in terms of compression, quality, and size.

Image Pre-Processing: Image Transformation, Geometric Transformations, Local pre-processing (Image smoothing, Edge detectors, Image Restoration, Canny edge detection, Line detection by local pre-processing operators), Image Filtering (Inverse Filtering, Wiener Filtering), Morphological Operations (on binary and grayscale images).

Activities:

- **Hands-on Lab:** Apply different filters (Gaussian, Median) and edge detectors (Sobel, Prewitt, Canny) to sample images using Python (OpenCV or skimage).

- **Assignment:** Implement line detection using the Hough Transform and analyze results.
- **Demonstration Task:** Restore a noisy or blurred image using Inverse and Wiener Filtering.

Image Segmentation: Local Image Segmentation: Thresholding, Edge-based segmentation, Region-based segmentation, Rule-based segmentation.
Segmentation by clustering pixels: basic clustering methods, The Watershed Algorithm, Segmentation Using K-means, Mean shift segmentation, Clustering and Segmentation with Mean Shift, Image Segmentation in Practice.

Activities:

- **Segmentation Assignment:** Segment a grayscale image using Otsu's thresholding, region growing, and edge-based methods.
- **K-means Segmentation:** Use OpenCV/Scikit-learn to segment an RGB image using K-means clustering.

Image Description and Feature Extractors: Texture features & Shape descriptors (Contours, Hu Moments), Color histograms and spatial color features, Harris Corner Detector, Scale-Invariant Feature Transform (SIFT),Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB),Feature maps and activation layers, Global Average Pooling vs. Flattening

Activities:

- **Feature Matching Project:** Detect and match keypoints between two images using SIFT and FLANN matcher.
- **Corner Detection Demo:** Implement Harris and Shi-Tomasi corner detectors and visualize differences.

Facet Model Recognition and Object Models: Definition and significance in object recognition,2.5D sketches and Marr's vision theory, Surface orientation, curvature, and shading models, Geometry-Based Object Representation, Volumetric (e.g., generalized cylinders, superquadrics) and surface-based models, Shape-from-shading and shape-from-motion approaches, Matching techniques for facet models, Use of graph matching and model indexing, Challenges in occlusion and illumination variations

Activities:

- **3D to 2D Matching Exercise:** Use available libraries (e.g., Open3D) to match a 3D model with 2D image features.
- **Discussion Task:** Short report on global vs. local features and where each would be used effectively (e.g., facial recognition vs. object tracking).

Applications of Computer Vision: Detecting Objects in Images, Object Recognition, Object Classification using Supervised learning algorithms Image segmentation and classification in radiology, Face detection and recognition Activity recognition and anomaly detection, Deep CNN Techniques.

Activities:

- **Object Detection Mini-Project:** Use a pre-trained YOLO or SSD model to detect and annotate objects in custom images or video.

- **Presentation:** Prepare a short presentation (5–10 min) explaining a real-world CV application (e.g., autonomous driving, face unlock, medical imaging) and how techniques from the course are used.

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		

References

1. Rafael Gonzalez, Richard E. Woods, “Digital Image Processing”, Fourth Edition, Pearson Education, 2018.
2. Forsyth D and Ponce J, Computer Vision A Modern Approach, Prentice Hall (2002).
3. Gonzalez, R. C., & Woods, R. E. (2017). Digital image processing (Global Ed., 4th ed.) Pearson Education Limited. ISBN 978-1-292-22304-9
4. Milman Sonka, Vaclav Halvac, Roger Boyle- Image Processing, Analysis and Machine Vision, 4th ed., 2015.
5. R Szeliski, Computer vision: algorithms and applications, Springer (2010)
6. R. O. Duda, P.E.Hart, and D.G.Stork,‖, Pattern Classification‖, 2nd edition, Springer, 2007.

E-Resources

1. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/
2. https://www.coursera.org/lecture/introduction-tensorflow/an-introduction-to-computer-vision-rGn1n
3. https://www.coursera.org/lecture/convolutional-neural-networks/edge-detection-example-4Trod
4. https://www.coursera.org/learn/computer-vision-basics
5. https://www.coursera.org/projects/computer-vision-object-detection
6. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and techniques of computer vision used for visual perception and image understanding.	--	--
CO2	**Analyze** computer vision algorithms to understand feature extraction, image representation, and performance evaluation.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** computer vision methods, models, and systems to assess their effectiveness for various visual analysis applications.	PO3 (2)	PSO2 (2)
CO4	**Design** computer vision solutions by selecting appropriate algorithms, models, and tools for real-world image and video processing tasks.	PO2 (1)	PSO1 (3)

IF25003	**Optimization Techniques**	L	T	P	C

		3	0	0	3

Course Objectives:

- Introduction to optimization techniques using both linear and non-linear programming.
- The focus of the course is on convex optimization though some techniques will be covered for non-convex function optimization too.
- After an adequate introduction to linear algebra and probability theory, students will learn to frame engineering minima maxima problems in the framework of optimization problems.

Introduction to Optimization: Introduction to Classical Methods & Linear Programming Problems Terminology, Design Variables, Constraints, Objective Function, Problem Formulation. Calculus method, Kuhn Tucker conditions, Method of Multipliers

Activities:

- **Case Study Analysis** – Identify design variables, constraints, and objective function in real-world engineering or business optimization scenarios.
- **Problem Formulation Exercise** – Formulate at least two optimization problems from different domains (e.g., manufacturing, logistics).

Linear Programming Problem: Linear Programming Problem, Simplex method, Two-phase method, Big-M method, duality, Integer linear Programming, Dynamic Programming, Sensitivity analysis.

Activities:

- **Duality Exploration** – Write the dual for given primal problems and interpret economic meanings.
- **Mini-Project** – Formulate and solve an Integer Linear Programming problem using any LP solver (e.g., Excel Solver, Python PuLP).

Single Variable Optimization Problems

Optimality Criterion, Bracketing Methods, Region Elimination Methods, Interval Halving Method, Fibonacci Search Method, Golden Section Method. Gradient Based Methods: Newton-Raphson Method, Bisection Method, Secant Method, Cubic search method.

Activities:

- **Comparative Analysis** – Compare performance (iterations, accuracy) of various methods for the same function.
- **Graphical Interpretation** – Plot objective functions and visualize the working of bracketing and gradient methods.
- **Quiz** – Objective-type questions on convergence criteria, optimality, and methods.

Multivariable and Constrained Optimization Techniques: Multi Variable and Constrained Optimization Technique, Optimality criteria, Direct search Method, Simplex search methods, Hooke-Jeeve‘s pattern search method, Powell‘s conjugate direction method, Gradient based method, Cauchy‘s Steepest descent method, Newton‘s method, Conjugate gradient method. Kuhn - Tucker conditions, Penalty Function, Concept of Lagrangian multiplier, Complex search method, Random search method.

Activities:

- **Hands-on Coding** – Implement the Cauchy’s Steepest Descent and Newton’s method for a two-variable function.
- **Simulation** – Demonstrate Powell’s and Hooke-Jeeves methods on a multivariable function using spreadsheets or code.
- **Mini Case Study** – Use Kuhn-Tucker conditions to solve a constrained multivariable problem.

Intelligent Optimization Techniques: Introduction to Intelligent Optimization, Genetic Algorithm: Types of reproduction operators, crossover & mutation, Simulated Annealing Algorithm, Particle Swarm Optimization (PSO), Genetic Algorithm, Bio-inspired algorithm: Particle Swarm Optimization (PSO) and Artificial Fish Swarm Algorithm (AFSA), differences between GA & GP, random population generation, solving differential equations using GP.

Activities:

- **Research Presentation** – Each group presents a real-world application of one intelligent technique (e.g., GA in scheduling, PSO in neural network training).
- **Differential Equation Solving** – Solve a simple ODE using genetic programming (symbolic regression).

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

E-resources:

1. https://www.nasa.gov/smallsat-institute/sst-soa/thermal-control/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and methods of optimization techniques used in engineering and computational problem solving.	--	--

CO2	**Analyze** optimization problems and solution techniques to understand constraints, objective functions, and algorithmic performance.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** optimization algorithms and approaches to assess their efficiency, accuracy, and applicability to real-world problems.	PO3 (2)	PSO2 (2)
CO4	**Design** optimal solutions by selecting and formulating appropriate optimization models and algorithms for practical applications.	PO2 (1)	PSO1 (3)

IF25004	**Cognitive Modelling**	L	T	P	C
		3	0	0	3

Course Objectives:

- To provide a comprehensive understanding of cognitive processes and computational models that simulate human cognition.
- The course aims to equip students with the ability to analyze, design, and implement cognitive models using AI and machine learning techniques. Students will gain practical exposure to cognitive architectures, decision-making models, and applications in human-computer interaction.

Cognitive Modelling: Fundamentals of cognition and cognitive science - Theories of cognition: Symbolic, Connectionist, and Hybrid approaches - Computational models of perception, attention, and memory.

Activities: Seminar on "Cognitive Architectures in AI Systems".

Machine Learning and Cognitive Systems: Neural networks and deep learning for cognitive modelling - Bayesian models of cognition and probabilistic reasoning - Reinforcement learning and decision-making models.

Activities: Flipped classroom with case studies on AI-driven cognitive models

Cognitive Architectures and Human-Computer Interaction: ACT-R, SOAR, and other cognitive architectures - Cognitive modelling in robotics and autonomous systems - Human-computer interaction and usability testing.

Activities: Project-based lab on implementing cognitive models in AI systems

Cognitive Modelling in Real-World Applications: Cognitive modelling in healthcare and neuroscience - Cognitive biases and their impact on AI decision-making - Ethical considerations in cognitive modelling.

Activities: Industrial/Field visit to AI research labs

Emerging Trends and Research Directions: Cognitive modelling in natural language processing - AI-driven creativity and problem-solving -Future directions in cognitive computing.

Activities: Reproduction of a recent IEEE/ACM research paper on cognitive modelling.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(iii). Activities: 10% (iv). Internal Theory Examinations:30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. John R. Anderson, Cognitive Psychology and Its Implications, 9th Edition, Worth Publishers, 2021.
2. David Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, MIT Press, 2010.
3. Randall C. O'Reilly and Yuko Munakata, Computational Explorations in Cognitive Neuroscience, MIT Press, 2000.
4. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2021.
5. Michael Dawson, Minds and Machines: Connectionism and Cognitive Science, Wiley, 2008.

E-resources:

1. NPTEL Course on Cognitive Science
2. Coursera: Computational Neuroscience
3. MIT OpenCourseWare: Cognitive Science
4. IEEE Xplore: Research Papers on Cognitive Modelling

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, theories, and approaches of cognitive modelling used to represent human cognition and behavior.	--	--
CO2	**Analyze** cognitive models and computational frameworks to understand perception, learning, memory, and decision-making processes.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** cognitive modelling techniques and architectures to assess their effectiveness in simulating human cognitive processes.	PO3 (2)	PSO2 (2)
CO4	**Design** cognitive models by selecting appropriate representations, algorithms, and validation methods for real-world cognitive systems.	PO2 (1)	PSO1 (3)

IF25005	**Natural Language Processing**	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the NLP Techniques
- To understand the tools for NLP
- To explore and gain a broad understanding of text data
- To understand the recent developments and emerging trends in the field of NLP

Overview and Morphology: N-grams Models of Syntax - Counting Words - Unsmoothed N-grams. Smoothing- Back-off Deleted Interpolation – Entropy - English Word Classes - Tag sets for English Part of Speech Tagging-Rule Based Part of Speech Tagging - Stochastic Part of Speech Tagging - Transformation-Based Tagging.

Activities :
N-gram Language Model Building:
Implement unigram and bigram models using a text corpus (e.g., movie reviews or news articles).
➤ Output: probability tables, sentence generation

Entropy Calculation:
Calculate the entropy of a given text using N-gram distributions.

Context Free Grammars:

Context Free Grammars for English Syntax- Context-Free Rules and Trees - Understand the network simulation tools. Sentence- Level Constructions–Agreement – Sub Categorization. Parsing – Top-down – Early Parsing -feature Structures – Probabilistic Context-Free Grammars.

Activities :

- **Parse Tree Drawing**:
  Use tools like NLTK or Stanford Parser to generate parse trees from input sentences.
- **Top-down vs. Earley Parsing**:
  Compare both techniques using a parser simulator or through code implementation.
- **Feature Structure Activity**:
  Implement unification-based grammar with feature structures in Prolog or Python.

Semantic Analysis

Representing Meaning-Meaning Structure of Language-First Order Predicate Calculus Representing Linguistically Relevant Concepts -Syntax-Driven Semantic Analysis - Semantic Attachments -Syntax-Driven Analyzer. Robust Analysis - Lexemes and Their Senses - Internal Structure - Word Sense Disambiguation - Information Retrieval.

Activities:

- **Semantic Parsing Exercise**:
  Create semantic trees or graphs for selected sentences with semantic role labeling.
- **Word Sense Disambiguation (WSD)**:
  Implement and evaluate Lesk algorithm or cosine similarity method using WordNet.
- **IR Task**:
  Build a basic search engine using TF-IDF for text documents with query matching.

Language Generation and Discourse Analysis

Discourse-Reference Resolution - Text Coherence -Discourse Structure – Coherence. Dialog and Conversational Agents - Dialog Acts – Interpretation - Conversational Agents. Language Generation–Architecture-Surface Realizations - Discourse Planning. Machine Translation -Transfer Metaphor– Interlingua – Statistical Approaches.

Activities :

- **Dialogue Act Classification**:
  Annotate or classify dialogue utterances based on dialog acts using a small conversation corpus.
- **Chatbot Simulation**:
  Build a simple rule-based or intent-based chatbot using Rasa, ChatterBot, or Dialogflow.
- **Discourse Coherence Evaluation**:
  Given multiple texts, evaluate their coherence manually and using automatic tools.

Natural Language Processing Models

Introduction – Models -and Algorithms - -Regular Expressions Basic Regular Expression Patterns – Finite State Automata Understand the wireless sensor network principles. Morphology -Inflectional Morphology - Derivational Morphology. Finite-State Morphological Parsing -- Porter Stemmer, statistical Models for Natural language processing. Large Language Model.

Activities :

- **Regex Practice Worksheet**:
  Create regex patterns for text extraction tasks (dates, emails, phone numbers, etc.).
- **Finite-State Automaton Simulation**:
  Design an FSA for recognizing valid identifiers or palindromes in English.
- **Morphological Analysis**:
  Build a morphological analyzer using affix rules (for inflectional and derivational morphology).

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i).Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. Daniel Jurafsky and James H Martin, ”Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, Prentice Hall, 2nd Edition, 2008
2. C. Manning and H. Schutze, “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA:,1999

3. TV Geetha, “Understanding Natural Language Processing (Machine Learning and Deep Learning Perspectives)” – Pearson 2024

E-resources:

1. NPTEL Applied NLP Course (Prof. Ramaseshan R)
2. NPTEL Deep Learning for NLP (Prof. Pawan Goyal)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and techniques of natural language processing for understanding and generating human language.	--	--
CO2	**Analyze** natural language processing algorithms to understand language representation, syntactic and semantic processing, and model performance.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** natural language processing methods and models to assess their effectiveness for various language-based applications.	PO3 (2)	PSO2 (2)
CO4	**Design** natural language processing solutions by selecting and integrating appropriate models, features, and tools for real-world language processing tasks.	PO2 (1)	PSO1 (3)

IF25006	**Data Science and Bioinformatics**	L	T	P	C
		3	0	0	3

Course Objective:

1. The course is aimed at providing a basic understanding to the students about bioinformatics methods and their in-depth applications for solving biological problems.
2. To be able to learn about the entire pipeline of a typical system involving data, collection, pre-processing, storage, retrieval, processing, analysis, and visualization.

Introduction to Data Science: Types of Data: Structured, Unstructured, Semi-structured, Descriptive Statistics & Inferential Statistics, Hypothesis Testing, t-tests, ANOVA, Data Cleaning and Preparation, Data Visualization Techniques (Bar, Pie, Histogram, Boxplot)

Activity:

- Data Visualization Mini Project:
- Students shall be given a real-world dataset (e.g., COVID-19, sales data).

Data Analysis and Manipulation: Data Acquisition and Preprocessing, Handling missing data, encoding categorical variables, Data cleaning strategies, Exploratory Data Analysis (EDA), Data Manipulation using Python & R, Time Series and Text Data Handling

Activity:

In pairs, students will perform EDA on a messy dataset (e.g., missing values, mixed data types).

Advanced Methods and Algorithms of Data Science: Need for advanced algorithms in modern datasets, Trade-offs: accuracy vs. complexity, Ensemble Methods, Optimization in Data Science, Advanced Clustering & Anomaly Detection.

Activity:

Students will implement and compare various models (e.g., decision trees, random forest, k-means, DBSCAN) on a dataset.

Bioinformatics: Importance and evolution, Classification of biological databases: Primary, secondary, and specialized, Overview of major sequence databases: GenBank, EMBL, DDBJ,Structural databases: PDB (Protein Data Bank), SCOP, CATH,Data formats: FASTA, GenBank, PDB format, Tools for database search and retrieval: Entrez, SRS, DBFetch, Current trends and challenges in data integration and curation.

Activity:

Students will be assigned **Database Exploration Task:** using different biological databases (e.g., GenBank, PDB, SCOP), Present a 5-slide summary of their findings and current trends.

Fundamentals of Sequence Analysis: Concepts of similarity, identity, and homology, Types of sequence alignment: global vs local, Dynamic programming algorithms: Needleman-Wunsch, Smith-Waterman, Heuristic algorithms: BLAST and FASTA – algorithms, parameters, interpretation of results, Gap penalties and alignment scoring schemes, Applications in function prediction and evolutionary studies

Activity:

Sequence Alignment Hands-on Session: Compare algorithms (Needleman-Wunsch vs Smith-Waterman), Interpret alignment scores and e-values, Predict potential gene/protein function based on alignment.

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		

References

1. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools. Cielen, Meysman,Ali. Dreamtech Press. ISBN-13: 978-9351199373.
2. Python Data Science Handbook: Essential Tools for Working with Data. Joel Grus. O'Reilly. ISBN-13: 978-9352134915
3. Grus, J. (2019). Data science from scratch: first principles with python. O'Reilly Media.
4. Cady, F. (2017). The data science handbook. John Wiley & Sons.
5. Singh, G. B. (2015). Fundamentals of bioinformatics and computational biology. Cham: Springer International Publishing, 159-170.
6. Introduction to Bioinformatics; Arthur M. Lesk; Oxford University Press, 2003
7. Bioinformatics: Methods and Applications Genomics, Proteomics, and Drug Discovery S.C. Rastogi, N. Mendiratta, P. Rastogi (3rd Edition) PHI Learning Private Limited New Delhi (2011)

E-Resources

Tools & Docs:

- Python Data Cleaning – Pandas Docs
- Matplotlib/Seaborn for Visualization
- R for Data Science – Online Book

Datasets:

- Kaggle Datasets
- UCI Machine Learning Repository

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, techniques, and applications of data science and bioinformatics in analyzing biological and biomedical data.	--	--
CO2	**Analyze** biological datasets using data science and bioinformatics methods to extract patterns, insights, and meaningful interpretations.	PO1 (3)	PSO1 (3)
CO3	Evaluate bioinformatics tools, data analysis techniques, and computational models to assess their effectiveness in biological research applications.	PO3 (2)	PSO2 (2)
CO4	Design data-driven bioinformatics solutions by selecting appropriate data science techniques, algorithms, and tools for real-world biological problems.	PO2 (1)	PSO1 (3)

		L	T	P	C
IF25007	**Deep Learning Techniques**	3	0	0	3

Course Objectives:

- To gain an in-depth understanding of the theoretical principles, algorithms, and methodologies underlying neural networks.
- To familiarize students and researchers with state-of-the-art deep learning techniques, methodologies, and optimization strategies aimed at enhancing model efficiency.
- To provide experiential learning in designing, training, and optimizing deep learning models for real-life applications.

Deep Learning Architectures: Deep feed forward networks: Gradient based learning – hidden units, Regularization: norm penalities – under constrained problems – dataset augmentation – noise robustness – sparse representations – bagging and other ensemble methods – drop out.

Activities: Seminar from research papers

Convolutional Networks: Convolution operation – sparse interactions – parameter sharing – pooling – variants of basic convolutional function – structured outputs – Convolutional neural networks (CNN) – CNN architecture – transfer learning architectures – AlexNet, ResNet, InceptioNet, MobileNet.

Activities: Project based learning: Deep learning model to classify images using custom CNN and pre-trained CNN, object detection using CNN, improving model performance by tuning the hyper parameters

Sequence Modelling: Recurrent neural networks – Bidirectional RNNs – encoder decoder sequence – to – sequence architectures – deep recurrent networks – recursive neural networks – LSTM – long term dependencies: challenges and optimization.

Activities: Project based learning**:** Sentiment analysis and time series forecasting using RNN

Deep Generative Models: Boltzmann machine, Restricted Boltzmann machine, deep belief network, deep Boltzmann machine, Generative adversarial networks, auto-encoders

Activities: Project based learning: Image generation using GAN

Explainability, Optimization, and Deployment: Visualizing Model Decisions, Explainable AI (XAI) Frameworks, Model Optimization Techniques, Adaptive Optimization Algorithms, Model Compression for Deployment, Deployment Tools and Pipelines

Activities: Optimize the model using early stopping and batch normalization, and deploy it using TensorFlow Lite (TFLite).

Weightage:	Continuous Assessment: 40%	End Semester Theory

	(iii). Activities: 10% (iv). Internal Theory Examinations: 30%	Examination: 60%

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. Ian Goodfellow, YoshuaBengio and Aaron Courville, " Deep Learning", MIT Press, 2017.
2. osh Patterson, Adam Gibson "Deep Learning: A Practitioner's Approach", O'Reilly Media, 2017
3. Giancarlo Zaccone, Md. RezaulKarim, Ahmed Menshawy "Deep Learning with TensorFlow: Explore neural networks with Python", Packt Publisher, 2017.
4. Umberto Michelucci "Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks" Apress, 2018.
5. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.

E-resources:

1. NPTEL – Deep Learning (IIT Kharagpur, Prof. Prabir Kumar Biswas)
2. NPTEL – Deep Learning Part 1 (IIT Madras, Profs. Khapra & Iyengar)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and learning mechanisms of deep learning techniques used in intelligent systems.	--	--
CO2	Analyze deep learning models and training algorithms to understand learning behavior, performance, and optimization challenges.	PO1 (3)	PSO1 (3)
CO3	Evaluate deep learning techniques, architectures, and training strategies to assess their effectiveness for complex data-driven applications.	PO3 (2)	PSO2 (2)
CO4	Design deep learning solutions by selecting appropriate models, architectures, and training methods for real-world problem scenarios.	PO2 (1)	PSO1 (3)

IF25008	Agile Methodology for Business Analysts	L	T	P	C
		3	0	0	3

Course Objectives:

- This course empowers students to master Agile methodologies, enabling them to translate business needs into technical value by applying critical analysis skills within modern software development lifecycles

Introduction Agile Philosophy and Values: Definition and origin of Agile-Agile vs traditional project management -The Agile Manifesto – values and principles-Three pillars of Agile: Transparency, Inspection, Adaptation-Agile mindset and cultural shift.

Activities:

- **Agile vs Traditional PM Debate**
  Split into two groups; one defends Agile, the other traditional project management. Compare benefits and limitations.
- **Agile Manifesto Deep Dive**
  Break into small teams and analyze each of the 4 values and 12 principles. Present real-world examples illustrating each.

Agile Frameworks and Methodologies: Overview of Agile frameworks: Scrum, Kanban, XP, SAFe - Roles and responsibilities: Product Owner, Scrum Master, Development Team - Agile ceremonies: Stand-ups, Reviews, Retrospectives - Tools used in Agile projects (e.g., Jira, Trello).

Activities:

- **Agile Ceremonies Practice**
  Conduct mock stand-ups, retrospectives, and sprint reviews using a fictional product backlog.
- **Tool Exploration**
  Hands-on session with Jira or Trello: create user stories, manage a sprint, and explore reporting features.

Business Analysis in Agile Environments: Evolving role of the Business Analyst in Agile - Requirement elicitation: User stories, Epics, Themes. Techniques: MoSCoW prioritization, INVEST criteria, Story Mapping, The Product Backlog and Definition of Ready/Done.

Activities:

- **User Story Writing Workshop**
  Practice writing effective user stories, epics, and themes based on a sample business case.
- **Story Mapping Exercise**
  Build a story map for a fictional product to understand user flow and identify MVP features.

- **MoSCoW Prioritization Game**
  Prioritize a list of features using MoSCoW. Justify your decisions based on business value and feasibility.

Systems, Service, and Lean Thinking for Agile BA: Systems thinking in business processes - Service design thinking and customer experience - Lean principles in Agile: Value stream mapping, eliminating waste - 8 Wastes of Lean and their application - POPIT model: People, Organization, Process, and Information & Technology.

Activities:

- **Customer Journey Mapping**
  Create a journey map to highlight pain points and design service improvements.
- **Systems Thinking Scenario**
  Analyze a complex system (e.g., customer support process) and identify interdependencies and feedback loops.

Agile Planning, Estimation, and Metrics: Release and iteration planning - Agile estimation techniques: Planning poker, T-shirt sizing - Agile metrics: Velocity, Burndown charts, Lead time, Cycle time - Scaling Agile: Challenges and success factors - Case studies and best practices in Agile project delivery.

Activities:

- **Agile Metrics Review**
  Match Agile metrics (e.g., Lead Time, Cycle Time) with project management challenges.
- **Case Study Review**
  Study a real-world Agile implementation. Identify what worked, what didn't, and why.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Highsmith, J. Agile Project Management: Creating Innovative Products, Addison-Wesley
2. BCS Learning and Development. Business Analysis (4th Edition)
3. Cohn, M. User Stories Applied: For Agile Software Development
4. Poppendieck, M. & Poppendieck, T. Lean Software Development
5. Lynda Girvan and Debra Paul Agile and Business Analysis BCS Learning and Development Ltd March 2017.

E-resources:

1. Agile Alliance: https://www.agilealliance.org
2. NPTEL: https://onlinecourses.nptel.ac.in/noc22_cs78/preview

3. Course era: https://www.coursera.org/learn/agile-atlassian-jira
4. IBM: https://www.edx.org/learn/agile/ibm-agile-and-scrum-fundamentals

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, values, and frameworks of agile methodology and their significance for business analysis.	--	--
CO2	Analyze agile processes and practices to understand requirement management, stakeholder engagement, and iterative development in business analysis.	PO1 (3)	PSO1 (3)
CO3	Evaluate agile models, project management techniques, and team practices to assess their effectiveness in delivering business value.	PO3 (2)	PSO2 (2)
CO4	Design business analysis workflows and solutions using agile principles by selecting appropriate frameworks, tools, and practices for real-world projects.	PO2 (1)	PSO1 (3)

IF25009	**Advanced Graphics and Animation**	L	T	P	C
		3	0	0	3

Course Objectives:

- Aims to provide an in-depth understanding of advanced computer graphics techniques, 3D modelling, rendering, and animation principles.
- Will learn real-time graphics programming, shader development, and physics-based animation using industry-standard tools.
- Emphasizes hands-on projects, research-oriented problem-solving, and emerging trends in AR/VR and GPU computing.

Core Graphics & Rendering: Graphics Pipeline Fundamentals – History and Evolution of Graphics APIs, OpenGL/Vulkan architecture, Vertex/fragment Shaders and their role in pipeline, Coordinate systems and transformations. Rendering Techniques – Physically Based Rendering (PBR) workflow, BRDF models and material systems, Shadow Mapping Techniques (PCF), Screen-Space Effects (SSAO). Global Illumination - Light Transport basics, Photon Mapping implementation, Real-time GI solutions.

Activities:

- Project – Implement a basic Ray Tracer
- Quiz – Rendering Equations & GPU Architecture

Practical Exercises:

- Implement a deferred renderer with multiple G-buffers
- Create a PBR material system with texture packing
- Compare Performance of Shadow Techniques

3D Modelling & Animation: Modelling Techniques - Subdivision surface modelling, Digital Sculpting Workflows, UV unwrapping and Texture Baking. Character Animation - Rigging fundamentals, Motion Capture Data Processing. Animation Retargeting Between Skeletons. Procedural Generation - Houdini basics, Terrain generation algorithms, Fractal-based Modelling.

Activities:

- Model Making – 3D Character Rigging
- Seminar – Compare Industry Tools like ZBrush Vs Mudbox

Practical Exercises:

- Create a fully rigged character with facial controls
- Process raw mocap data into usuable animation clips
- Generate a procedural environment using Houdini

Shaders & GPU Programming: Shader Programming - GLSL/HLSL Essentials, Shader Debugging Techniques, Computer Shader Fundamentals, GPU Memory Architecture. Visual Effects – Volumetric Rendering, Water Simulation Shaders, Particle Systems on GPU, Post-Processing Stack. Optimization - Shader LOD Systems, Instruction-Level Optimization, Async Compute Scheduling, Vulkan/DXR Performance Tuning.

Activities:

- Flipped Classroom – Debug a Fragment Shader
- Poster Presentation – Visual Effects Breakdown

Practical Exercises:

- Implement a full-screen CRT Distortion Effect
- Create a GPU-based Particle System
- Optimize an existing shader by 30%

AR/VR & Real-Time Graphics: VR Fundamentals - Stereoscopic Rendering Math, Lens Distortion Correction, Motion controllers. AR Development - SLAM Algorithms, Marker-based Vs Markless AR, Plane Detection and Occlusion, ARCore/ARKit Integration. Performance Optimization – Draw Call Batching Strategies, Level of Detail Systems, Occlusion culling Techniques, Multithreading Rendering.

Activities:

- Industrial Visit – AR/VR Startup or Gaming Studio
- Project-based Learning – Build a VR walkthrough

Practical Exercises:

- Build a VR Scene with Hand Tracking
- Create an AR furniture Placement App

- Profile and Optimize a Unity Scene

Emerging Trends

Neural Rendering – Neural Radiance Fields (NeRF), GAN-based Texture Synthesis, Style transfer for art direction, AI-Assisted Animation. Advanced Simulation - Fluid Dynamics on GPU, Cloth and Hair Simulation, Crowd Simulation AI. Ethical considerations - Deepfake Detection, Digital Likeness Rights, Bias in Generative AI.

Activities:

- Reproduction of Research Paper – Implement a SIGGRAPH paper
- Debate – Topic “Ethics of AI-generated Animation”

Practical Exercises:

- Train a NeRF Model for a Simple Object
- Implement a Basic Fluid Simulation
- Analyze Ethical Case Studies

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(v). Activities: 10% (vi). Projects: 20% (vii).Presentations:10% (viii). Internal Theory Examinations: 10%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Jason Gregory, “Game Engine Architecture”, Third Edition, CRC Press, 2018.
2. John M. Blain, “The Complete Guide to Blender Graphics”, Seventh Edition, CRC Press, 2022.
3. Matt Pharr, Wenzel Jakob, and Greg Humphreys, “Physically Based Rendering”, Fourth Edition, Morgan Kaufmann, 2023.
4. Tomas Akenine-Möller, Eric Haines, Naty Hoffman, Angelo Pesce, Michal Iwanicki, and Sebastien Hillaire, “Real-Time Rendering”, Fourth Edition, CRC Press, 2018.
5. Steve Marschner and Peter Shirley, “Fundamentals of Computer Graphics”, Fifth Edition, AK Peters, 2021.

E-resources:

1. https://nptel.ac.in/courses/106102063
2. https://nptel.ac.in/courses/106103224
3. https://nptel.ac.in/courses/106106090
4. Unity Learn (Real-Time Development): https://learn.unity.com/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, techniques, and tools used in advanced computer graphics and animation for creating interactive visual content.	--	--
CO2	Analyze graphics and animation algorithms, rendering techniques, and modeling methods to understand their computational and visual performance.	PO1 (3)	PSO1 (3)
CO3	Evaluate advanced graphics and animation approaches, frameworks, and tools to assess their effectiveness in producing realistic and efficient visualizations.	PO3 (2)	PSO2 (2)
CO4	Design interactive graphics and animation solutions by selecting appropriate algorithms, tools, and techniques for real-world multimedia applications.	PO2 (1)	PSO1 (3)

CP25C17	**Edge and Fog Computing**	L	T	P	C
		3	0	0	3

Course Objective:

This course aims to provide students with a comprehensive understanding of designing computing systems that process data closer to where it's created, rather than sending everything to distant cloud servers. Students will learn to build smart applications that work across different computing layers - from powerful cloud servers to small edge devices like sensors and smartphones. By the end, students can create practical solutions for smart cities, healthcare, and industry that respond quickly and work efficiently.

Cloud to Edge Computing Paradigms: Evolution from Cloud to Fog and Edge Computing - IoT Architecture Requirements and Latency Limitations - Fog vs Edge vs Cloud: Differences in Computation and Storage - Multi-access Edge Computing (MEC) Concepts - Multi-tier Architectural Models for Distributed Systems

Activities:

- Case Study: IoT solution using fog or edge
- Group Discussion: Business opportunities in edge markets

Edge Offloading and System Optimization: Edge Offloading Strategies and Placement Issues - Challenges in Latency, Energy Efficiency, QoS, and Privacy - Queuing Theory and System Performance Metrics - Optimization Techniques using Machine Learning for Slice Orchestration

Activities:

- Performance Comparison: Cloud vs Edge
- Mathematical Modeling: Latency or offloading simulation

Resource Federation and Network Slicing: Cloud-to-Fog-to-Things (C2F2T) Integration - Network Slicing for 5G/6G: Slice Types and Lifecycle - Resource Virtualization: SDN, NFV, and Container-Based Federation - Modeling and Simulation for Resource Management

Activities:

- Implementation Case Study: 5G network slicing for verticals
- Quiz: Virtualization and resource slicing concepts

Middleware for Edge Systems: Middleware Design Goals: Heterogeneity, Discovery, Scalability – Service Orchestration at the Edge - Lightweight Containers: Docker and Kubernetes on Constrained Devices - Edge Cluster Formation and Node Coordination - Communication Protocols: MQTT, CoAP, and Microservices - Distributed Storage, Synchronization, and Fault Recovery

Activities:

- Prototype: Middleware architecture on edge devices
- Containerization Project: Docker/Kubernetes deployment
- Security Analysis: Vulnerabilities in edge middleware

Intelligence and Analytics at the Edge: Data Management at the Fog Layer: Caching, Placement, Lifecycle - Federated Learning and Distributed ML for Edge Analytics - Stream Processing and Real-Time Analytics Frameworks - Use Cases: Smart Cities, Healthcare, Smart Homes - Security Challenges: ML-Based Threat Detection, Behavioral Analysis - Privacy Techniques: Differential Privacy, Federated Anonymization

Activities:

- Implement Predictive Analytics: Apache Kafka + Flink
- Develop ML Model: IoT threat detection
- Case Study: Privacy-preserving fog computing

Advanced Technologies for Secure and Efficient Edge Computing: Blockchain for Fog Resource Security and Transparency - Software-defined Edge Networks and Intent-Based Networking - Green and Sustainable Edge Computing-Security and Privacy Challenges in Edge Environments

Activities:

- Case Study: Edge security or sustainability
- Technical Report: Designing a secure, green edge architecture

Future Directions and Emerging Trends: Edge AI and TinyML: Neural Compression, Quantization, Model Deployment - 6G Vision: Ultra-Low Latency, Terahertz Communications - Advanced Paradigms: Digital Twins, Serverless Edge, Cognitive Networks

Activities:

- Research Project: Emerging tech integration with edge computing
- Presentation: Future research directions and innovation trends
- Technical presentation on future research directions in edge computing

Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

References:

1. Rajiv Pandey, Sunil Kumar Khatri, Neeraj Kumar Singh, Parul Verma, Artificial Intelligence and Machine Learning for EDGE Computing, 1st Edition,2022, Academic Press
2. S. Goundar, Ed., ‘Edge Computing - Technology, Management and Integration’. IntechOpen,2023.
3. Ahmed Banafa, Secure and Smart Internet of Things (loT) using Blockchain and Artificial Intelligence (AI).2018, River publications
4. Javid Taheri, Schahram Dustdar, Albert Zomaya, Shuiguang Deng, Edge Intelligence from Theory to Practice, 2023, Springer
5. Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications Deepak Gupta, Aditya Khamparia ,2021, Wiley-IEEE Press

E-Resources:

1. Kubernetes on Edge: K3s Documentation and Tutorials - https://k3s.io/
2. Foundation of Cloud IoT Edge ML, NPTEL, Prof. Rajiv Misra, IIT Patna - https://archive.nptel.ac.in/courses/106/104/106104242/
3. Open5GS Documentation - https://open5gs.org/
4. Eclipse IoT Working Group - https://iot.eclipse.org/
5. TensorFlow Lite for Microcontrollers
6. https://www.tensorflow.org/lite/microcontrollers
7. Apache Flink Training - https://flink.apache.org/
8. Hyperledger Fabric Tutorials - https://hyperledger-fabric.readthedocs.io/

   Subedi, P., Alsadoon, A., Prasad, P.W.C. et al. Network slicing: a next generation 5G perspective. J Wireless Com Network 2021, 102 (2021). https://doi.org/10.1186/s13638-021-01983-7

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and principles of edge and fog computing for distributed and low-latency applications.	--	--
CO2	Analyze edge and fog computing frameworks, data processing models, and resource management strategies to understand performance, scalability, and efficiency.	PO1 (3)	PSO1 (3)

CO3	Evaluate edge and fog computing solutions, protocols, and deployment strategies to assess their effectiveness for real-time and distributed applications.	PO3 (2)	PSO2 (2)
CO4	**Design** edge and fog computing systems by selecting appropriate architectures, computation models, and communication strategies for real-world applications.	PO2 (1)	PSO1 (3)

IF25C03	**DevOps and Micro Services**	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the DevOps culture and its significance in modern software development.
- Explore automation tools and practices used in continuous integration, delivery, and deployment, gain hands-on experience with version control, build automation, testing, and infrastructure as code and develop the ability to monitor, manage, and secure DevOps pipelines effectively.

DevOps Foundations and Process Automation

Evolution of Software Development Life Cycle (SDLC), Agile, Lean, and DevOps What is DevOps? Principles and Practices, DevOps vs Traditional IT, Benefits and Challenges of DevOps, DevOps Lifecycle and Key Concepts (CI/CD, Infrastructure as Code, Monitoring, Feedback Loops), DevOps Toolchain Overview. DevOps maturity and automation

Activities :

- **Quiz:** Short quiz on DevOps principles, benefits, and challenges.
- **Case Study:** Analyze a company’s software delivery before and after adopting DevOps. Identify key improvements.

Version Control and Continuous Integration: Basics: Repositories, Branching, Merging, Rebase, GitHub/GitLab for Collaboration, Continuous Integration Concepts, Build Automation Tools: Maven, Gradle, CI Tools: Jenkins, GitHub Actions, GitLab CI, Creating and Managing CI Pipelines.

Activities:

- **Hands-on Lab:** Initialize a Git repository, create branches, and perform merges and rebases.
- **Collaboration Exercise:** Use GitHub/GitLab to fork a repository, submit pull requests, and review code.

Configuration Management and Containerization: Configuration Management Tools: Ansible, Puppet, Chef, Infrastructure as Code (IaC) Concepts, Introduction to Docker: Images, Containers, Dockerfiles, Docker Compose, Container Orchestration Overview (Kubernetes basics), Comparing VMs vs Containers

Activities:

- **Tool Demo:** Write a basic Ansible playbook or Puppet manifest to configure a web server.
- **Discussion:** Compare VM and container architectures; list pros and cons.

Continuous Deployment and Monitoring: Continuous Deployment vs Continuous Delivery, Tools for Deployment: Jenkins Pipelines, Spinnaker, Monitoring Tools: Prometheus, Grafana, ELK Stack, Logging and Alerting Mechanisms, DevOps Metrics and KPIs.

Activities:

- **Monitoring Setup:** Install and configure Prometheus and Grafana dashboards to monitor application metrics.
- **Log Analysis:** Set up ELK stack to collect and analyze logs from a demo app.

Cloud, Security, and Best Practices: Introduction to Cloud Platforms: AWS, Azure, GCP, DevOps on Cloud: CloudFormation, Terraform, SecDevOps: Integrating Security into DevOps, Blue-Green and Canary Deployments, Best Practices in DevOps: Culture, Collaboration, Feedback Loops.

Activities:

- **Security Review:** Conduct a security checklist for CI/CD pipeline (identify vulnerabilities and propose fixes).
- **Deployment Strategy Roleplay:** Simulate blue-green and canary deployments and discuss rollback strategies.
- **Culture Brainstorm:** Develop a plan to improve collaboration and feedback loops in a team environment.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(ix). Activities: 10% (x). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, The DevOps Handbook, IT Revolution Press, 2016.
2. Jez Humble and David Farley, Continuous Delivery, Addison-Wesley, 2010.
3. Kelsey Hightower et al., Kubernetes: Up and Running, O’Reilly Media.
4. Official documentation of tools like Git, Docker, Kubernetes, Jenkins, Ansible, Terraform.

E-resources:

NPTEL Video

1. https://youtu.be/2N-59wUIPVI?si=4SrHAJvOsOKmxi21
2. https://youtu.be/87ZqwoFeO88?si=084GDdu-pDfsVKMq

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and practices of DevOps and microservices architecture for efficient software delivery.	--	--
CO2	**Analyze** DevOps pipelines and microservices-based systems to understand deployment strategies, integration, and scalability.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** DevOps tools, microservices frameworks, and operational practices to assess their effectiveness in continuous delivery and system reliability.	PO3 (2)	PSO2 (2)
CO4	**Design** DevOps-enabled microservices solutions by selecting appropriate architectures, tools, and workflows for real-world software applications.	PO2 (1)	PSO1 (3)

IF25010	**Design Thinking and Patterns**	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the principles and process of design thinking in engineering and problem solving.
- To provide knowledge about identifying user needs and developing innovative solutions.
- To explore design patterns in software/system/product design.
- To foster empathy, creativity, ideation, and prototyping skills. To enable the application of reusable design patterns in system design and development

Design Thinking: Design Thinking – Definition, Importance, Scope - The Design Thinking Process: Empathize, Define, Ideate, Prototype, Test - Divergent vs. Convergent Thinking - The Role of Creativity and Innovation - Real-world applications in Engineering and Technology - Mindset of a Design. Thinker.

Activities:

- **Divergent vs. Convergent Thinking Exercise**: Given a problem, students list as many solutions as possible (divergent), then narrow down to the best 3 (convergent).
- **Mindset Reflection:** Write a short journal entry on what being a design thinker means and how creativity and innovation influence problem-solving.
- **Case Study Discussion:** Analyze a real-world engineering or technology innovation that used design thinking and discuss the phases involved.

Empathy and Problem Definition: Techniques to Gain Empathy: User Interviews, Surveys, Observations- Persona Creation and User Journey Mapping - Defining Problems: Framing, Reframing and Problem Statements - Point-of-View Problem Framing - Tools: Empathy Map, Need Statements.

Activities:

- **Empathy Map Workshop:** Fill in empathy maps to capture user feelings, needs, and pain points.
- **Problem Statement Framing:** Practice writing problem statements using Point-of-View framing and need statements from collected data.

Ideation and Prototyping: Ideation Techniques: Brainstorming, Scamper, Mind Mapping - Selecting Ideas – Feasibility, Viability, Desirability - Prototyping – Types and Methods - Low-Fidelity vs High-Fidelity Prototypes - Storyboarding and Scenario Building - Feedback and Iterative Design.

Activities:

- **Brainstorming Session:** In groups, brainstorm as many ideas as possible for a given challenge using SCAMPER and mind mapping techniques.
- **Idea Selection Matrix:** Evaluate ideas based on feasibility, viability, and desirability to select the best one.
- **Prototype Creation:** Build low-fidelity prototypes (paper sketches, cardboard models) for the selected idea.

Design Patterns and Classification: Introduction to Design Patterns: Definition and Advantages - Pattern Catalogs: Gang of Four (GoF) Patterns - Types of Patterns: Creational, Structural, Behavioral - Pattern Format and Documentation - Applying Patterns in Software and System Design.

Activities:

- **Pattern Documentation Exercise:** Practice documenting a chosen design pattern using the standard pattern format.
- **Pattern Classification Quiz:** Match different patterns to their categories (Creational, Structural, Behavioral).
- **Group Discussion:** Analyze the advantages of applying specific design patterns in software/system design problems.

Pattern Application and Case Studies: Use Case Driven Pattern Selection - Refactoring and Improving Legacy Systems with Patterns - Anti-Patterns and Best Practices - Real-time Applications and Case Studies: UI/UX, Product Design, Software Systems - Group Project/Case Presentation on Design Thinking Process.

Activities:

- **Use Case Analysis:** Given a use case, select and justify design patterns that can be applied to solve the problem.
- **Refactoring Challenge:** Refactor a provided legacy code snippet or system design by applying suitable design patterns.
- **Anti-Patterns Identification:** Identify anti-patterns in given scenarios and suggest best practices to avoid them.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Tim Brown, Change by Design, Harper Business, 2009.
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
3. Jeanne Liedtka, Tim Ogilvie, Designing for Growth: A Design Thinking Tool Kit for Managers, Columbia Business School Publishing, 2011.
4. Nigel Cross, Design Thinking: Understanding How Designers Think and Work, Berg, 2011.
5. Donald A. Norman, The Design of Everyday Things, Basic Books, Revised Edition, 2013.
6. Alan Cooper, About Face: The Essentials of Interaction Design, Wiley, 2014.
7. Brown, Tim. (2009). *Change by Design: How Design Thinking Creates New Alternatives for Business and Society*, Harvard Business Press.
8. IDEO.org. (2015). *The Field Guide to Human-Centered Design*, IDEO.org. (2015).

E-resources:

NPTEL Video

1. https://youtu.be/2N-59wUIPVI?si=4SrHAJvOsOKmxi21
2. https://youtu.be/87ZqwoFeO88?si=084GDdu-pDfsVKMq

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and methodologies of design thinking and design patterns in software development.	--	--
CO2	**Analyze** design problems and software patterns to understand their applicability, efficiency, and impact on system architecture.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** design thinking approaches and pattern-based solutions to assess their effectiveness in creating user-centric and maintainable software.	PO3 (2)	PSO2 (2)
CO4	**Design** agentic AI solutions by selecting appropriate agent architectures, learning methods, and coordination mechanisms for real-world applications.	PO2 (1)	PSO1 (3)

IF25011	Information and Cyber Security	L	T	P	C
		3	0	0	3

Course Objectives:

- To impart comprehensive knowledge on the principles and practices of information and cyber security.
- The course aims to develop competency in identifying vulnerabilities, applying appropriate security solutions, and ensuring data protection in real-world systems. Students will gain practical exposure to tools, current threats, and techniques relevant to cyber defence.

Information Security and Threat Landscape: Basic concepts: Security goals, attacks, services, mechanisms -Threat models and vulnerabilities - Cryptography fundamentals (symmetric and asymmetric techniques).

Activities:

- **Wireshark** – Analyze plaintext vs. encrypted traffic to show confidentiality/integrity principles.

Network and Web Security: Secure communication protocols (SSL/TLS, IPSec) - Firewalls and Intrusion Detection Systems (IDS) - Network vulnerabilities and mitigation techniques - Web application security (SQL injection, XSS, CSRF).

Activities:

- Detect Port Scanning with IDS using tool Snort.
- Perform SQL injection on login form using tool DVWA

System Security and Malware Analysis:OS security principles, Secure boot, Access control models - Malware types, detection techniques, and sandboxing - Anti-virus and endpoint security mechanisms.

Activities:

- Project-based lab on malware analysis using open-source tools.

Cyber Security in Practice: Cybercrime and legal frameworks (IT Act, GDPR overview) - Security in cloud computing and IoT - Digital forensics and incident response.

Activities:

- Industrial/Field visit to Security Operations Center (SOC).

Emerging Trends and Ethical Aspects: Zero trust architecture, blockchain security, AI in security - Cyber ethics, privacy concerns, social engineering.

Activities:

- Simulating a Social Engineering Attack using Open-Source Intelligence (OSINT)
- Reproduction of a recent IEEE/ACM security research paper.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i).Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. William Stallings, Cryptography and Network Security, 8th Edition, Pearson, 2023.
2. Michael E. Whitman and Herbert J. Mattord, Principles of Information Security, 7th Edition, Cengage, 2021.
3. Mark Stamp, Information Security: Principles and Practice, 3rd Edition, Wiley, 2021.
4. Charlie Kaufman, Radia Perlman, and Mike Speciner, Network Security: Private Communication in a Public World, 2nd Edition, Pearson, 2022.
5. Nina Godbole and Sunit Belapure, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley, 2020.

E-resources:

1. NPTEL Course on Cyber Security
2. Coursera: IBM Cybersecurity Analyst Professional Certificate
3. Cybrary Security Resources
4. OWASP Web Security Resources
5. Kali Tools – Penetration Testing Suite

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and practices of information and cyber security in modern computing systems.	--	--
CO2	Analyze security threats, vulnerabilities, and protective measures to understand risks and mitigation strategies in information systems.	PO1 (3)	PSO1 (3)
CO3	Evaluate security frameworks, cryptographic techniques, and risk management approaches to assess their effectiveness in safeguarding information assets.	PO3 (2)	PSO2 (2)
CO4	Design secure systems and protocols by selecting appropriate security mechanisms, policies, and risk management strategies for real-world applications.	PO2 (1)	PSO1 (3)

IF25012	**Geo Spatial Informatics**	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the physical principles of remote sensing and sensor technologies and develop skills in digital image processing for thematic mapping and analysis.
- Providing an understanding of GIS concepts, spatial data structures, and analysis techniques.
- Familiarize satellite navigation systems and develop the ability to create and serve spatial data over the web.

Remote Sensing and Image Processing: Fundamentals of remote sensing-satellite sensors: optical, thermal, microwave, LiDAR-Radiometric and Geometric-Image Classification: Supervised, unsupervised-Image enhancement and transformation techniques.

Activities: Quiz, Google Earth Engine image processing exercises.

Geographic Information Systems (GIS):GIS basics: vector/raster models, coordinate systems- Data input methods: GPS, digitizing, remote sensing-Spatial analysis: buffering, overlay, interpolation-Topology creation and editing-Cartographic principles and map layout design-Metadata standards and spatial data infrastructure (SDI)-Applications in utilities, transport, resource planning.

Activities: Presentation on GIS operations using QGIS/ArcGIS with vector/raster data.

GNSS and Geodesy: Introduction to GNSS: GPS, GLONASS, IRNSS - Signal structure, errors, atmospheric effects - Geodesy: ellipsoid, geoid, datum, projections - Map projections and coordinate conversions - Photogrammetry: stereo vision, DEM generation.

Activities: Quiz and geodetic computation exercises.

Spatial Databases & Web GIS: Database fundamentals and spatial extensions - Spatial indexing and query optimization - PostGIS and SQL for spatial data - Web GIS architectures - Open-source tools: GeoServer, Leaflet.

Activities: Reproduction of Research Paper.

Machine Learning in Geospatial Analytics: Introduction to ML & spatial data types - Supervised/unsupervised learning - Deep learning (CNN) for image classification - Time series forecasting and spatio-temporal modelling - Accuracy assessment and model validation - Object detection in remote sensing imagery.

Activities:

- Project based learning on AI/ML applications in agriculture, disaster, urban analysis.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Thomas M. Lillesand, Ralph W. Kiefer, Jonathan W. Chipman, "Remote Sensing and Image Interpretation", 7th edition, Wiley, Hoboken NJ, USA, 2015.
2. John R. Jensen, "Introductory Digital Image Processing: A Remote Sensing Perspective", 4th edition, Pearson, Upper Saddle River NJ, USA, April 21, 2015.
3. Chor Pang Lo, Albert K. W. Yeung, "Concepts and Techniques of Geographic Information Systems", 2nd edition—Prentice Hall, Upper Saddle River NJ, USA, 2002.
4. Elliott D. Kaplan, Christopher Hegarty, "Understanding GPS/GNSS: Principles and Applications", 3rd edition, Artech House, London/Boston, 2017.
5. Pinde Fu, Jiulin Sun, "Web GIS: Principles and Applications", 1st edition, Esri Press, Redlands CA, USA, 2010.
6. Mikhail Kanevski, Alexei Pozdnoukhov, Vadim Timonin, "Machine Learning for Spatial Environmental Data: Theory, Applications, and Software", 1st edition, EPFL Press (Taylor & Francis/CRC), New York/London, 2009.

E-resources:

1. NPTEL – Geographic Information Systems (IIT Roorkee, Prof. Arun Saraf)
2. NPTEL – Spatial Informatics (IIT Kharagpur, Prof. Soumya K. Ghosh)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and applications of geospatial informatics in spatial data analysis and geographic information systems.	--	--
CO2	Analyze geospatial data, mapping techniques, and spatial analysis methods to understand patterns, relationships, and trends.	PO1 (3)	PSO1 (3)
CO3	Evaluate geospatial tools, algorithms, and methodologies to assess their effectiveness in solving real-world spatial problems.	PO3 (2)	PSO2 (2)
CO4	Design geospatial solutions by selecting appropriate data models, analytical methods, and visualization techniques for practical applications.	PO2 (1)	PSO1 (3)

IF25C04	Block Chain Technologies	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the foundational principles of blockchain technology.
- To explore the architecture and consensus mechanisms of distributed ledgers.
- To design and develop smart contracts and decentralized applications.
- To evaluate real-world use cases and security issues in blockchain networks.

Blockchain and Distributed Ledger Technologies: Centralized vs. Decentralized systems- Blockchain fundamentals: blocks, chains, hash functions, Merkle trees-Types of blockchain: Public, Private, Consortium-Consensus mechanisms: PoW, PoS, PBFT-Key challenges: scalability, privacy, interoperability.

Activities: Analyze and compare centralized vs decentralized systems (e.g., traditional banking vs Bitcoin).

Cryptographic Foundations and Consensus: Cryptographic hash functions-Digital signatures and public-key cryptography-Mining and proof-of-work in Bitcoin-Byzantine fault tolerance and proof-of-stake mechanisms-Tokenomics and incentive structures.

Activities: Simulate a blockchain manually with students acting as nodes. Include hash functions, blocks, and consensus.

Smart Contracts and Ethereum Platform: Ethereum architecture and EVM-Solidity programming language-Smart contract design and development-Gas optimization and security best practices-Deployment and interaction with contracts.

Activities:

- Teams debate PoW vs PoS vs PBFT – focusing on efficiency, security, and real-world applications.

Blockchain Platforms and Applications: Hyperledger Fabric and enterprise blockchains-Corda and permissioned networks-Decentralized Finance (DeFi) and NFTs-Blockchain in supply chain, healthcare, identity, and voting-Interoperability solutions (Polkadot, Cosmos).

Activities: Given a list of real-world use cases (e.g., Bitcoin, Hyperledger, Ripple), identify the type of blockchain used and justify the classification.

Challenges, Trends, and Research Directions: Security and privacy in blockchain systems-Legal, regulatory, and ethical considerations-Scalability solutions: Layer-1 vs Layer-2-Blockchain and IoT Integration-Future trends: zk-SNARKs, DAGs, CBDCs.

Activities: Create a visual representation of block structure, Merkle tree, and hash chaining.

Weightage:	Continuous Assessment: 40%	End Semester Theory

	(i). Activities: 10% (ii). Internal Theory Examinations: 30%	Examination: 60%

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2016.
2. Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020.
3. Andreas M. Antonopoulos, Mastering Bitcoin, O'Reilly Media
4. Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media

E-resources:

1. NPTEL – Blockchain and Its Applications (IIT Kharagpur)
2. NPTEL – Introduction to Blockchain Technology and Applications (IIT Kanpur)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and principles of blockchain technologies and their applications in decentralized systems.	--	--
CO2	Analyze blockchain protocols, consensus mechanisms, and distributed ledger models to understand security, scalability, and performance.	PO1 (3)	PSO1 (3)
CO3	Evaluate blockchain platforms, smart contract frameworks, and deployment strategies to assess their effectiveness for real-world applications.	PO3 (2)	PSO2 (2)
CO4	Design blockchain-based solutions by selecting appropriate architectures, consensus protocols, and smart contract models for practical use cases.	PO2 (1)	PSO1 (3)

IF25013	**Climate Change and Disaster Management**	L	T	P	C
		3	0	0	3

Course Objectives:

- The course aims to provide an in-depth understanding of climate change science, its impact on disasters, and mitigation strategies.
- It equips students with analytical and practical skills to assess climate risks, develop disaster resilience plans, and implement sustainable solutions.
- It integrates interdisciplinary approaches, including policy frameworks, technological innovations, and community-based adaptation methods.

Fundamentals of Climate Change: Earth's Climate System - Greenhouse Effect – Radiative Forcing – Natural Vs Anthropogenic drivers of climate change – Global Temperature Rise – Sea-Level Rise – Ocean Acidification. Global & Regional Climate Trends: IPCC Report (2023) findings – Regional Climate Variability – Climate Change Hotspots in India. Climate Policies & Frameworks: India's Nationally Determined Contributions – Paris Agreement – Kyoto Protocol.

Activities:

- **Seminar:** Presenting key findings from IPCC report.
- **Quiz:** MCQs on climate science fundamentals.
- **Flipped Classroom:** Debate on "Anthropogenic vs. Natural Climate Change".

Climate-Induced Disasters: Hydrometerological Disasters: Floods – Cyclones - Droughts & Water Scarcity. Extreme Weather Events: Heatwaves - Glacial Lake Outbursts Flood – Wildfires. Case Studies & Risk Assessment: Comparative Analysis of Disasters in India Vs. Global – Climate Risk Mapping Using GIS Tools.

Activities:

- **Field Visit:** Assessment of Disaster-prone area.
- **Model Making:** Simulate a cyclone trajectory using meteorological data.
- **Reproduction of Research Paper:** Critique a study on GLOF risks.

Disaster Risk Reduction (DRR) & Management: Early Warning Systems: IMD's Cyclone Forecasting – Tsunami warning Systems – AI-based Disaster Prediction Models. Resilient Infrastructure: Flood-resistant Urban Planning - Earthquake-resistant buildings - Smart cities & climate-proofing. Community-Based DRR: Role of NGOs - Indigenous Knowledge in Disaster Preparedness - Gender-Sensitive Disaster Response.

Activities:

- **Poster Presentation:** Design an early warning system for landslides.
- **Industrial Visit:** NDMA (National Disaster Management Authority) or state disaster response centres.
- **Role-Play:** Simulate a community DRR meeting.

Mitigation & Adaptation Strategies: Renewable Energy & Decarbonization: Solar/Wind Energy Transitions - Carbon Capture & Storage (CCS) Technologies - Green Hydrogen Economy. Nature-Based Solutions: Mangrove Restoration – Urban Green Spaces – Agroforestry for Climate Resilience. Climate-Smart Agriculture: Drought-resistant Crops – Precision Farming & IoT in Agriculture – Policy Frameworks.

Activities:

- **Project-Based Learning:** Develop a carbon-neutral plan for a village.
- **Guest Lecture:** Expert talk on renewable energy policies.
- **Quiz:** MCQ test on mitigation technologies.

Policy, Governance & Future Trends: Paris Agreement Compliance Mechanisms – India's National Action Plan on Climate Change (NAPCC) – State Action Plans (SAPCCs). AI & Big Data in Disaster Prediction – Blockchain for Climate Finance Transparency – Remote Sensing & GIS Applications. SDG 13 (Climate Action) Linkages with Other Goals – Corporate Climate Responsibility – Youth & Public Participation in Climate Advocacy.

Activities:

- **Group Project:** Draft a mock climate policy for a state.
- **Research Paper Review:** Analyze a recent journal article on AI in disaster management.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(i). Quizzes (10%), (ii). Assignments (10%), (iii). Seminars/Posters (10%), (iv). Field Reports (10%)	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

Internal Examinations: TWO tests

References:

1. IPCC, Climate Change 2023: Synthesis Report, Cambridge University Press, 2023.
2. Krishna S. Vatsa, "Disaster Management", Pearson, 2022.
3. Rajib Shaw & Fuad Mallick, "Climate Change Adaptation and Disaster Risk Reduction: An Asian Perspective", Emerald Publishing, 2021.
4. Saleemul Huq, Jeffrey Chow, Adrian Fenton, Clare Stott, Julia Taub and Helen Wright "Confronting Climate Change in Bangladesh", Springer, 2020.
5. Walter Leal Filho, "Handbook of Climate Change Adaptation", Springer, 2022.

E-resources:

1. https://nptel.ac.in/courses/120108558
2. https://nptel.ac.in/courses/105106707
3. https://nptel.ac.in/courses/124107010
4. NASA Climate Change (Data & Reports)- https://science.nasa.gov/climate-change/
5. UNFCCC (Climate Policy Resources) - https://unfccc.int/
6. The Atlas of Climate Change (University of California Press) by Kirstin Dow and Thomas E. Downing https://ncse.ngo/files/pub/evolution/excerpt--atlas.pdf

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, causes, and impacts of climate change and the principles of disaster management.	--	--
CO2	Analyze climate change data, hazard patterns, and disaster risks to understand vulnerabilities and mitigation strategies.	PO1 (3)	PSO1 (3)
CO3	Evaluate climate adaptation measures, disaster management frameworks, and risk reduction strategies to assess their effectiveness in protecting communities and environments.	PO3 (2)	PSO2 (2)
CO4	Design disaster preparedness and climate adaptation plans by selecting appropriate tools, strategies, and technologies for real-world applications.	PO2 (1)	PSO1 (3)

IF25014	**Financial Data Analytics**	L	T	P	C
		3	0	0	3

Course Objectives:

- This course aims to equip students with the knowledge and practical skills required to analyze financial data using statistical, machine learning, and time series techniques.
- It emphasizes hands-on experience with Python-based tools for financial forecasting, risk analysis, and portfolio management.
- It also fosters critical thinking in applying analytics to real-world financial decision-making scenario.

Foundations of Financial Data and the Analyst's Toolkit: Basics of Financial Markets (Equity, Fixed Income, Derivatives). Types of Financial Data: Market Data (OHLCV), Fundamental Data (from financial statements), Economic Data (e.g., CPI, GDP), and Alternative Data. Characteristics of Financial Data: Time-series nature, non-stationarity, volatility clustering, fat tails. Setting up the Python Environment: Anaconda, Jupyter Notebooks/Lab. Essential Libraries: pandas for data manipulation, numpy for numerical operations.

Activities:

- Identify and categorize different types of financial data (market, fundamental, economic, alternative).
- Create a comparison table showing sample data from each category

Exploratory Financial Data Analysis (EFDA) and Visualization: Calculating Financial Returns: Simple, Logarithmic, and Cumulative Returns. Measuring Risk: Standard Deviation (Volatility), Variance, and Downside Deviation. Risk-Adjusted Returns: Sharpe Ratio, Sortino Ratio. Visualizing Financial Data: Line charts for prices, histograms for return distributions, candlestick charts. Analyzing Relationships: Correlation and Covariance matrices, heatmap visualization, scatter plots. Rolling Statistics: Simple Moving Averages (SMA), Exponential Moving Averages (EMA), and rolling volatility.

Activities:

Visual Analysis:

- Plot candlestick charts using plotly or mplfinance.
- Create histograms for return distributions.
- Generate correlation matrix heatmaps using seaborn.

Time Series Analysis for Financial Data: Core Concepts of Time Series: Stationarity, Autocorrelation (ACF), and Partial Autocorrelation (PACF). Decomposition of Time Series: Trend, Seasonality, and Residuals. Forecasting with ARIMA Models: Autoregressive (AR), Moving Average (MA), and integrated (I) components. Modeling Volatility: Introduction to ARCH and GARCH models to capture volatility clustering.

Activities:

- Use rolling mean and ADF (Augmented Dickey-Fuller) test to check for stationarity in financial time series.
- Decompose a time series (e.g., airline passenger data or stock prices) into trend, seasonality, and residuals using seasonal_decompose.
- Fit ARIMA models to forecast stock prices or economic indicators.

Quantitative Modeling and Machine Learning in Finance: Factor Models: Capital Asset Pricing Model (CAPM) and Fama-French models. Implementing Factor Models using Linear Regression. Supervised Learning for Prediction: Regression - Linear Regression or Gradient Boosting (XGBoost) to predict future returns based on technical indicators. Classification - Logistic Regression or SVM to predict market direction (up/down). Unsupervised Learning for Pattern Recognition- Clustering (K-Means): Grouping stocks into clusters based on their risk-return profiles.

Activities:

Market Direction Classification:

- Use Logistic Regression or SVM to classify whether a stock’s price will go up or down.
- Evaluate model with confusion matrix, accuracy, precision, recall.

Portfolio Management and Risk Analysis:

Modern Portfolio Theory (MPT): The trade-off between risk and return.Mean-Variance Optimization: Calculating the Efficient Frontier and the Optimal Risky Portfolio. Risk Management Techniques: Value at Risk (VaR): Historical Method, Parametric (Variance-Covariance) Method. Monte Carlo Simulation for risk analysis.

Activities:

Monte Carlo Simulation:

- Simulate multiple future price paths for a portfolio using Geometric Brownian Motion.
- Estimate risk measures from simulations (e.g., max drawdown, worst-case loss).

Weightage:	Continuous Assessment: 40% (i). Time series forecasting for stock prices or interest rates (10%) (ii). Group assignment: Credit risk model for loan applications (10%) (iii). End-to-end analysis using real-world financial datasets. (10%)	End Semester Theory Examination: 60%

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Yves Hilpisch "Python for Finance: Mastering Data-Driven Finance" O'Reilly Media
2. Richard A. DeFusco, Dennis W. McLeavey, Jerald E. Pinto, David E. Runkle, Quantitative Investment Analysis" (CFA Institute Investment Series), Wiley.
3. Marcos López de Prado "Machine Learning for Asset Managers" Cambridge University Press
4. David Ruppert, "Statistics and Data Analysis for Financial Engineering", Springer
5. Salih N. Neftci, "Principles of Financial Engineering", Academic Press.

E-resources:

1. Kaggle Datasets: https://www.kaggle.com/datasets?tags=NaN-finance
2. NPTEL: https://onlinecourses.nptel.ac.in/noc21_ma39/preview
3. Course era: https://www.coursera.org/specializations/investment-management-python-machine-learning
4. Python Institute: https://pypi.org/project/yfinance/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, methods, and applications of financial data analytics in analyzing financial and economic data.	--	--
CO2	Analyze financial datasets using analytical and statistical techniques to understand trends, risks, and investment patterns.	PO1 (3)	PSO1 (3)
CO3	Evaluate financial analytics models, tools, and forecasting techniques to assess their effectiveness in decision-making processes.	PO3 (2)	PSO2 (2)
CO4	Design financial data analytics solutions by selecting appropriate models, algorithms, and tools for real-world financial applications.	PO2 (1)	PSO1 (3)

IF25015	Ethical Hacking and Management	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand ethical hacking concepts, tools, and techniques.
- Apply penetration testing methods systematically.
- Analyze security vulnerabilities in systems and networks.
- Manage security operations within ethical and legal frameworks.

Introduction to Ethical Hacking: Introduction to cybersecurity threats and vulnerabilities - Definition and scope of ethical hacking - Types of hackers (white, gray, black hats) - Hacker mindset and skills - Legal and ethical implications (Information Technology Act, GDPR, HIPAA) - Cyber kill chain & MITRE ATT&CK framework

Activities:

- **Ethical Hacker Role Play:** Students assume the roles of white, gray, and black hat hackers and debate legal/ethical scenarios.
- **Cyber Laws Quiz:** A quiz covering IT Act, GDPR, HIPAA, and relevant laws.
- **Kill Chain Mapping Exercise:** Map a known attack (e.g., phishing > malware > exfiltration) to the Cyber Kill Chain and MITRE ATT&CK.

Foot printing, Reconnaissance, and Social Engineering: Passive vs Active Reconnaissance - WHOIS, DNS Interrogation - Social Engineering Techniques: Phishing, Baiting, Pretexting - Email harvesting and metadata analysis - Information Gathering tools: OSINT With Buscador

Activities:

- **Social Engineering Simulation:** Design a phishing email or pretexting script (in a controlled lab environment).
- **Metadata Analysis:** Analyze the metadata of sample files (e.g., images, documents) to uncover hidden information.
- **Mini Project:** Conduct passive reconnaissance on a fictitious company using Recon-ng and document findings.

Scanning, Enumeration, and Vulnerability Analysis: Network scanning techniques (ping sweep, port scanning)- Enumeration of systems and services (SNMP, NetBIOS, LDAP) - Vulnerability scanning tools: Nessus, OpenVAS, Nikto - CVE, CVSS, and exploit databases - Web application vulnerabilities (OWASP Top 10).

Activities:

- **Network Scanning Lab:** Use tools like Nmap and Zenmap to perform ping sweeps and port scans on a sandbox network.
- **Enumeration Lab:** Enumerate services using tools like SNMPwalk, enum4linux, and ldapsearch on a test environment.

System Hacking, Exploits, and Malware: Password cracking techniques (rainbow tables, hashcat, John the Ripper) - Privilege escalation and maintaining access -

Trojans, rootkits, keyloggers - Exploit development basics: buffer overflows, shellcode - Metasploit Framework and Kali Linux tools.

Activities:

- **Privilege Escalation Lab:** Practice basic Linux/Windows privilege escalation on vulnerable VMs (like Metasploitable).
- **Malware Analysis Demo:** Analyze a harmless Trojan or keylogger in a sandbox and document behavior.

Management of Cybersecurity, Risk & Compliance:

Risk management principles and frameworks (ISO 27001, NIST RMF) - Security audits, logs, and incident response - Role of a Chief Information Security Officer (CISO)- Business continuity and disaster recovery - Cybersecurity policy, compliance, and governance.

Activities:

- **Audit and Log Review Lab:** Analyze logs (e.g., Apache, Windows Event Viewer) for suspicious activity.
- **CISO Interview or Role Simulation:** Interview a cybersecurity leader (virtually or in person) or simulate the CISO role in a crisis scenario.

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(iv). Activities: 10% (v). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. EC-Council. Certified Ethical Hacker (CEH) Version 12 Official Courseware. EC-Council, 2022.
2. G. Weidman, Penetration Testing: A Hands-On Introduction to Hacking. San Francisco, CA: No Starch Press, 2014..
3. J. Erickson, Hacking: The Art of Exploitation, 2nd ed. San Francisco, CA: No Starch Press, 2008.
4. R. E. Smith, Elementary Information Security, 2nd ed. Burlington, MA: Jones & Bartlett Learning, 2015.

E-resources:

1. Cyber Security Tools Techniques and Counter Measures By Prof. Dr. Nilesh K Modi Dr. Babasaheb Ambedkar Open University, Ahmedabad, Gujrat

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and ethical considerations of ethical hacking and security management in information systems.	--	--
CO2	Analyze system vulnerabilities, attack techniques, and security assessment methods to understand risks and threat mitigation strategies.	PO1 (3)	PSO1 (3)
CO3	Evaluate ethical hacking tools, security frameworks, and management practices to assess their effectiveness in protecting organizational assets.	PO3 (2)	PSO2 (2)
CO4	Design secure systems and security management strategies by selecting appropriate ethical hacking techniques, controls, and policies for real-world environments.	PO2 (1)	PSO1 (3)

IF25C05	**Generative AI**	L	T	P	C
		3	0	0	3

Course Objectives:

- Introduce the fundamental concepts and evolution of generative modeling in AI.
- Study deep generative models including VAEs and GANs and their applications.
- Explore transformer-based models for multimodal generation.
- Provide hands-on experience in building generative AI systems using open-source tools.
- Analyze ethical, legal, and deployment aspects of generative technologies.

Generative AI: Overview of Generative AI – Differences between Discriminative and Generative Models – Applications in Art, Code, and Text – Overview of Deep Learning – Introduction to Latent Variable Models – Statistical Foundations – Random Sampling – Maximum Likelihood Estimation – Brief Introduction to AI Ethics.

Activities:

- **Group Discussion**: Compare and contrast discriminative vs. generative models with real-world examples.
- **Mini Research Presentation**: Each student presents one real-world application of Generative AI (art, music, code, etc.).
- **Concept Map**: Create a visual map showing the relationship between deep learning, latent variables, and generative models.

Autoencoders and Variational Methods: Autoencoders – Undercomplete vs. Overcomplete AEs – Denoising Autoencoders – Variational Autoencoders (VAE) – Latent Space Sampling – Kullback-Leibler Divergence – Reparameterization Trick – ELBO Objective – Applications: Face Generation, Image Compression.

Activities:

- **Build a Basic Autoencoder**: Use PyTorch or TensorFlow to implement an undercomplete autoencoder on the MNIST dataset.
- **Denoising Experiment**: Corrupt input images and train a **denoising autoencoder**
- **Conceptual Quiz**: Short quiz on latent space, VAE architecture, and reparameterization trick.

Generative Adversarial Networks (Gans): Architecture – Generator and Discriminator – Minimax Objective – Deep Convolutional GAN (DCGAN) – Conditional GANs – CycleGAN, StyleGAN – Challenges: Mode Collapse, Non-convergence – Evaluation Metrics: Inception Score, FID.

Activities :

- **Architecture Sketching**: Students draw and annotate the GAN architecture showing interactions between the Generator and Discriminator.
- **Experiment with DCGAN**: Modify a pre-trained DCGAN and generate new images.
- **Case Study Discussion**: Compare CycleGAN vs. StyleGAN for different use cases (e.g., image translation vs. high-quality image generation).

FOUNDATION MODELS AND MULTIMODAL GENERATION

Sequence Modeling – Attention Mechanism – Transformer Architecture – GPT Models – Text-to-Text and Text-to-Image Generation – CLIP and DALL·E – Multimodal Datasets – Prompt Engineering – Role of RLHF in Alignment.

Activities:

- **Transformer Demo**: Use Hugging Face to explore transformer-based models (e.g., GPT-2, T5).
- **Prompt Engineering Challenge**: Students experiment with different prompts using ChatGPT or similar to generate desired outputs.
- **Text-to-Image Generation**: Generate images using **DALL·E mini** or **Craiyon**, and critique results.

Ethics, Safety and Deployment: Generative AI Risks – Deepfakes and Detection – Bias and Hallucinations – Model Interpretability – Copyright and Intellectual Property – Safety Layers and Human Feedback – Open Source vs. Commercial APIs – Tools for Deployment – Hugging Face, Replicate, ONNX, Gradio.

Activities :

- **Workshop on Detection Tools**: Use available tools to detect AI-generated content (e.g., Deepfake Detection).
- **Group Presentation**: Propose safety mechanisms for deploying a generative AI model in a sensitive domain (healthcare, law, etc.).

Capstone Project and Industry Applications: Case Studies in Healthcare, Gaming, EdTech, Robotics – Model Compression and Optimization – Transfer Learning in

Generative Context – Custom Fine-tuning – Deployment on Cloud Platforms – API Integration – Real-time Generative Systems.

Activities :

Understand real-world uses of generative AI across industries.

- Assign each team/individual one industry (e.g., healthcare, gaming).
- Research a real-world application using generative AI (e.g., AI-assisted medical report generation, generative design in games).
- Prepare a **5–7-minute presentation** covering:
    - Problem solved
    - Type of generative model used
    - Results and challenges
    - Ethical implications

Weightage:	Continuous Assessment: 40%	End Semester Theory Examination: 60%
	(iii). Activities: 10% (iv). Internal Theory Examinations: 30%	

Mandated Activities with marks:

Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
2. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media, 2nd Edition, 2022.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.
4. Sebastian Raschka and Vahid Mirjalili, Python Machine Learning, Packt Publishing, 3rd Edition, 2020.
5. Max Tegmark, Life 3.0: Being Human in the Age of Artificial Intelligence, Penguin Books, 2017.

E-resources:

1. https://huggingface.co/learn
2. https://www.coursera.org/learn/build-basic-generative-adversarial-networks-gans
3. https://openai.com/research
4. https://developers.google.com/machine-learning/guides/text-classification
5. https://nptel.ac.in/courses/106105216 – Deep Learning by Prof. Mitesh Khapra (IITM)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and applications of generative artificial intelligence in creating data-driven content and solutions.	--	--
CO2	Analyze generative AI models and architectures to understand data generation processes, learning behavior, and performance characteristics.	PO1 (3)	PSO1 (3)

CO3	Evaluate generative AI techniques, models, and ethical considerations to assess their effectiveness and impact in real-world applications.	PO3 (2)	PSO2 (2)
CO4	Design generative AI solutions by selecting appropriate models, architectures, and training strategies for practical content generation tasks.	PO2 (1)	PSO1 (3)