Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus of

B.Tech. (AI & DS)

Effective from Academic Year 2026-27

Prepared by: Board of Studies in AI & DS

Approved by: Academic Board, Vishwakarma Institute of Technology, Pune

Chairman–BOS **Chairman–Academic Board**

Institute Vision

"To be globally acclaimed Institute in Technical Education and Research for holistic Socio- economic development".

Institute Mission

- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.
- Promote Research Culture among Students and Faculty through Projects and Consultancy.
- To make students Socially Responsible Citizen.

Department Vision

"To offer quality academic environment with the modern infrastructure to cater the demand of AI and DS careers with the research aptitude"

Department Mission

- To promote employability and entrepreneurship skills among students in the AI-DS and IT domains.
- To impart quality education with the focus on design, development and analysis using Interdisciplinary approach.
- To encourage students-faculty participation in research and development in collaboration with industry.
- To prepare students for solving problems of societal benefits and make them responsible citizens

Knowledge and Attitude Profile (WK)

WK	WK Statements
WK1	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK2	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK3	A systematic, theory-based formulation of engineering fundamentals required in the Engineering discipline.
WK4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK5	Knowledge, including efficient resource use, environmental impacts, whole-life cost, reuse of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
WK6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK7	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK9	Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

List of Programme Outcomes (PO)(Applicable for T.Y and Final Year)

Graduates will be able

PO	PO Statement
PO1	**Engineering knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified inWK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	**Design/development of solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4	**Conduct investigations of complex problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles And commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and teamwork:** Function effectively as an individual, and as a member or Leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply the set one's own work, as a member and leader in a team to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to Engage in independent and life-long learning in the broadest context of technological change.

List of Programme Outcomes (PO)(Applicable for S.Y)

Graduates will be able

PO	PO Statement
PO1	**Engineering knowledge:** Apply knowledge of mathematics, natural sciences, engineering fundamentals, and an engineering specialization to develop solutions for complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems using first principles of mathematics, natural sciences, and engineering sciences to reach substantiated conclusions.
PO3	**Design/Development of solutions:** Design solutions for complex engineering problems and design systems, components, or processes that meet specified needs with due consideration for public health and safety, whole-life cost, net-zero carbon, culture, society, and the environment.
PO4	**Conduct investigations of complex problems:** Conduct investigations of complex engineering problems using research-based knowledge and methods, including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO5	**Engineering Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, recognizing their limitations.
PO6	**The Engineer and the World:** Analyze and evaluate societal, health, safety, legal, cultural, environmental, economic, and sustainability aspects relevant to engineering practice, and understand the responsibilities of professional engineers in contributing to sustainable development.
PO7	**Ethics:** Apply ethical principles and commit to professional ethics, responsibilities, inclusivity, equity, and norms of engineering practice.
PO8	**Individual and Collaborative Team Work:** Function effectively as an individual, and as a member or leader in diverse, multidisciplinary, and inclusive teams.
PO9	**Communication:** Communicate effectively on complex engineering activities with the engineering community and society through reports, documentation, presentations, and by giving and receiving clear instructions.
PO10	**Project Management and Finance:** Apply engineering and management principles and economic decision-making to manage projects in multidisciplinary environments.
PO11	**Life-Long Learning:** Recognize the need for and engage in independent and life-long learning to adapt to technological advancements and changing professional practices.

Programme Specific Outcomes (PSO)

PSO	PSO Statement
PSO1	Graduates will be able to identify, enhance, and validate algorithms that use intelligence to make sensible judgments in real-world situations by applying their programming skills.
PSO2	Graduates will be able to design applications in the fields of data mining, artificial intelligence, and associated fields.

Program Educational Objectives (PEO)

PEO	PEO Focus	PEO Statement
PEO1	Preparation	Graduates will be equipped with a strong foundation in mathematics, computational principles, and engineering practices to prepare for advanced studies, research, or successful careers in artificial intelligence, data science, and allied fields.
PEO2	Core competence	Graduates will develop expertise in core areas of artificial intelligence and data science, including machine learning, deep learning, big data analytics, and computational intelligence, enabling them to design and implement effective solutions to real-world challenges.
PEO3	Breadth	Graduates will acquire multidisciplinary knowledge by integrating concepts from engineering, natural sciences, social sciences, and management, enabling them to work on innovative and holistic solutions across diverse domains.
PEO4	Professionalism	Graduates will demonstrate ethical practices, leadership qualities, effective communication skills, and a commitment to teamwork while addressing societal and industrial challenges responsibly and sustainably.
PEO5	Learning Environment	Graduates will thrive in a learning environment that promotes curiosity, innovation, and continuous professional development, equipping them with the ability to adapt to technological advancements and pursue lifelong learning.

B.Tech. Artificial Intelligence & Data Science (applicable w.e.f. AY 26-27)
Index

Nomenclature for Teaching and Examination Assessment Scheme AY 2026-27

Sr No.	Category	Head of Teaching/ Assessment	Abbreviation used
1	Teaching	Theory	Th
2	Teaching	Laboratory	Lab
3	Teaching	Tutorial	Tut
4	Teaching	Open Elective	OE
5	Teaching	Multi-Disciplinary	MD
6	Teaching	Artificial Intelligence & Data Science	AI&DS
7	Assessment	Laboratory Continuous Assessment	CA
8	Assessment	Mid Semester Assessment	MSA
9	Assessment	End Semester Assessment	ESE
10	Assessment	Home Assignment	HA
11	Assessment	Course Project	CP
12	Assessment	Group Discussion	GD
13	Assessment	PowerPoint Presentation	PPT
14	Assessment	Class Test –1	CT1
15	Assessment	Class Test –2	CT2
16	Assessment	Mid Semester Examination	MSE
17	Assessment	End Semester Examination	ESE
18	Assessment	Written Examination	WRT
19	Assessment	Multiple Choice Questions	MCQ
20	Assessment	Laboratory	LAB

Title: Course Structure **SEM-I** **FF No.653**
Branch: AI&DS Year: SY AY: 2026-27 Module: III&IV

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											Credits
						Examination Scheme and Marks										Total	
			Theory	Lab	Tut	Lab CA	T1	MSE	T2	ESE Pract +CVV	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI2301	Data Structures	3	2	0	-	-	-	-	40	40	-	-	-	20	100	4
S2	AI2302	Foundations of Intelligent Digital Systems	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S3	AI2303	Database Management Systems	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S4	AI2304	Discrete Structures and Computational Theory	2	2	0	10	-	40(W)	-	-	-	-	20	-	30	100	3
S5	MM0105/ MM0106/ MM 0107	MDM(ARVR, IP, CG)	2	0	1	-	-	40(W)	-	-	-	-	20	20	20	100	3
S6	HS2002	From Campus to Corporate	2	0	0	-	-	50(R)	-	-	-	50(R)	-	-	-	100	2
S7	HS2001	Reasoning and Aptitude Development-3	-	-	-	-	-	-	-	-	-	-	-	-	-	100	1
S8	AI2305	Design Thinking- III	-	-	1	-	-	-	-	-	-	-	-	-	-	100	1
S9	AI2306	Engineering Design & Innovation – III	-	4	-	-	-	30	-	-	-	70	-	-	-	100	2
Total			12	12	2	30	70	80	70	40	80	120	120	-	120	900	22

Title: Course Structure **SEM-II** **FF No.653**
Branch: AI&DS Year: SY AY: 2026-27 Module: III&IV

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											Credits
						Examination Scheme and Marks										Total	
			Theory	Lab	Tut	Lab CA	T1	MSE	T2	ESE Pract +CVV	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI2307	Object Oriented Programming	3	2	0	-	-	-	-	40	40	-	-	-	20	100	4
S2	AI2308	Computer Network	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S3	AI2309	Mathematical foundations for Artificial Intelligence	2	2	0	10	-	40(W)	-	-	-	-	20	-	30	100	3
S4	AI2310	Probability and Applied Statistics	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S5	MM0105/ MM0106/ MM0107	MDM(ARVR, IP, CG)	2	0	1	-	-	40(W)	-	-	-	-	20	20	20	100	3
S6	HS2003	From Campus to Corporate	2	0	0	-	-	50(R)	-	-	-	50(R)	-	-	-	100	2
S7	HS2004	Reasoning and Aptitude Development -3	-	-	-	-	-	-	-	-	-	-	-	-	-	100	1
S8	AI2311	Design Thinking- IV	-	-	1	-	-	-	-	-	-	-	-	-	-	100	1
S9	AI2312	Engineering Design & Innovation – IV	-	4	-	-	-	30	-	-	-	70	-	-	-	100	2
Total			12	12	2	30	70	80	70	40	80	120	120	-	120	900	22

Title: Course Structure **SEM-I** **FF No.653**

Branch: AI&DS Year: TY AY: 2026-27 Module:V and VI

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											
						Examination Scheme and Marks										Total	Credits
			Theory	Lab	Tut	Lab CA	T1	MSE	T2	ESE Pract	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI3201	Artificial Intelligence	3	2	0	-	-	-	-	40	40	-	20	-		100	4
S2	AI3202	Machine Learning	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S3	AI3206A/ AI3206B	Operating System / System Programming /MLops/Basics of Game Development	3	2	0	-	-	-	-		-	-	20	20+20	40	100	4
S4	AI04COURSERA	PEC 2 :Coursera	1	0	0											100	3
S5	AI05COURSERA	OE 1 : Coursera	1	0	-	-	-	-	-	-						100	4
S6	MM 0108A/MM 0108B	MDM 3 Blockchain/ IoT	2	0	1	-	-		-	-	40		-	20+10	30	100	3
S7	AI 3204	Engineering Design & Innovation – V	-	-	-	-	-	30	-	-	-	70	-	-	-	100	1
S8	AI3205	Design Thinking- V	-	-	1	-	-	-	-	-	-	-	-	-	-	100	2
Total																900	24

Audit Courses for Third Year:

1. Mainframe Technologies
2. Basics of Game Development

Title: Course Structure **SEM-II** **FF No.653**

Branch: AI&DS Year: TY AY: 2026-27 Module: V and VI

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											
						Examination Scheme and Marks										Total	Credits
			Theory	Lab	Tut	Lab CA	T1	MSE	T2	ESE Pract	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI 3010	Deep Learning	3	2	0	-	-	-	-	40	40	-	20	-		100	4
S2	AI3011	Complexity Algorithm	2	2	0	10	-	-	-	-	40	-	20	-	30	100	3
S3	AI3207A/ AI3207B/ AI3207C	PEC 3 : Quantum Computing/Federated Learning/ Software Design and Methodologies	3	2	0	-	-	-	-		-	-	20	20+20	40	100	4
S4	AI04COURSERA	PEC 4: Coursera	1	0	0											100	3
S5	AI05COURSERA	OE 2 : Coursera	1	0	-	-	-	-	-	-						100	4
S6	MM0109A /MM0109 B	MDM 4 : Edge AI / NLP	2	0	1	-	-		-	-	40		-	20+10	30	100	3
S7	AI3205	Engineering Design & Innovation VI	-	-	-	-	-	30	-	-	-	70	-	-	-	100	1
S8	AI3206	Design Thinking- VI	-	-	1	-	-	-	-	-	-	-	-	-	-	100	2
Total																	24

TY Coursera Course List:

Sr No	Course Code	Name
1	MD3102	Meta Back-End Developer
2	MD3106	Microsoft Power BI Data Analyst
3	MD3112	Google Cybersecurity
4	MD3113	Google Data Analytics
5	MD3118	IBM Data Science
6	MD3119	Fractal Data Science
7	MD3121	IBM DevOps and Software Engineering
8	MD3141	AWS Cloud Technology Consultant
9	MD3101	IBM Full Stack Software Developer
10	MD3147	Deep Learning.ai Deep Learning

TY Open Elective Bucket

Sr No	Course Code	Name
1	MD3201	Bucket 1:FinTech
2	MD3202	Bucket 2:Digital Marketing
3	MD3203	Bucket 3:Entrepreneurship
4	MD3204	Bucket 4:Management

Title: Course Structure **FF No.653**
Branch: AI&DS **Year: B. Tech AY: 2026-27** **Module: VII (Department Module)**

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme							Total	Credits
						CA		MSA		ESA				
			Theory	Lab	Tut	Lab	HA	MSE	PPT	ESE	GD	Viva		
OE1	LL*	LinkedIn Learning*	-	-	-	-	-	-	-	100	-	-	100	2
OE2	AI4025/ AI4015/	High Performance Computing/ Network Security	2	-	-	-	10	30	-	30	-	30	100	2
OE3	AI4007/ AI4028	Reinforcement Learning/ Introduction to Large Language Models	2	-	-	-	10	30	-	30	-	30	100	2
	AI4005	Major Project	-	20	-	-	-	30	-	70	-	-	100	9
	AI4023	Design Thinking-VII	-	-	-	-	-	-	-	100	-	-	100	1
Total			4	20	-	-	20	90	-	330	-	60	500	16

Module: VIII (Internship Module)

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme							Total	Credits
						CA		MSA		ESA				
			Theory	Lab	Tut	Lab	HA	MSE	PPT	ESE	GD	Viva		
S1	AI4008	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	15
S2	AI4009	Project Internship	-	32	-	-	-	30	-	70	-	-	100	15
S3	AI4011	International Internship	-	32	-	-	-	30	-	70	-	-	100	15
S4	AI4010	Research Internship	-	32	-	-	-	30	-	70	-	-	100	15
S5	AI4023	Design Thinking - VII	-	-	-	-	-	-	-	100	-	-	100	1
		Total	-	32	-	-	-	30	-	310	-	-	400	16

LinkedIn Learning Courses*

Subject Code		Subject Name
MD4274	Bucket 1	Large Language Models Skill Development
MD4275		Mastering Microsoft Power BI
MD4276		Generative AI Skills for Developers
MD4277		Career in Data Analysis
MD4278		Concepts of Data Visualization and Storytelling
MD4279		AWS Certified Solutions Architect
MD4280		IT Security Specialist
MD4281		Technical Program Management
MD4282	Bucket 2	Natural Language Processing Skill Development
MD4283		Prompt Engineering Skills
MD4284		Essentials in Generative AI
MD4285		Python in Finance
MD4286		Understanding Quantum Computing
MD4287		Foundational Maths for Machine Learning

S. Y. B. Tech. Artificial Intelligence and Data Science

AY 2026-27

FFNo.: 654

AI2301: Data Structures

Teaching Scheme	Examination Scheme
Credits: 4	CP: 20 Marks
Lectures : 3 Hrs/week	PR+CVV: 40 Marks
Practical : 2 Hrs/week	ESE: 40 Marks

Course Prerequisites:

1. Basic knowledge of programming using C/C++ or any structured programming language.
2. Understanding of variables, data types, operators, expressions, and control structures.
3. Familiarity with functions, recursion, arrays, pointers, and basic file handling.
4. Fundamental knowledge of algorithmic thinking and problem-solving techniques.
5. Basic understanding of computer organization and memory concepts.

Course Objectives:

1. To understand the fundamental concepts, classifications, and memory representations of linear and non-linear data structures.
2. To develop the ability to select and implement appropriate data structures for solving computational problems efficiently.
3. To analyze the time and space complexity of various data structures and their associated algorithms.
4. To apply linear data structures such as arrays, linked lists, stacks, and queues in designing solutions for real-world applications.
5. To implement and perform operations on non-linear data structures including trees, graphs, heaps, and hashing techniques.
6. To design efficient software solutions by integrating suitable data structures and algorithms for problem-solving in Artificial Intelligence, Data Science, and other computing applications.

Course Relevance:

Data Structures is a fundamental course for Computer Science, Artificial Intelligence, and Data Science students. It develops the ability to organize, store, and manage data efficiently using appropriate linear and non-linear data structures. The course enhances problem-solving skills by enabling students to select suitable data structures and

algorithms for different computational problems while considering time and space complexity. The concepts learned in this course form the foundation for advanced subjects such as Algorithms, Database Management Systems, Operating Systems, Computer Networks, Artificial Intelligence, Machine Learning, and Big Data Analytics. The knowledge of data structures is highly relevant in software development, system design, data engineering, and research, making it an essential skill for both industry and higher studies.

SECTION-I

Arrays, Linked Lists, Sorting ,Searching, Stack, Queue, Time & Space Complexity

Unit 1: Arrays, Linked List, Time and Space Complexity Analysis **(7 Hours)**

Arrays: Single and Multidimensional arrays

Linked Lists: Dynamic memory allocation, Singly Linked Lists, doubly linked Lists, Circular liked lists, and Generalized linked lists, skip list, Applications of Linked list.

Time & Space Complexity Analysis: Asymptotic Notations, Types of Problems,: P, NP, NP-Hard, NP Complete, Examples.

Unit 2: Sorting and Searching Techniques **(7 Hours)**

Sorting Techniques: Insertion, Selection, Bubble, Bucket, Merge, Quick,Radix sort, and heap sort.

Search techniques: Binary search, Fibonacci search.

Unit 3: Stack and Queue **(7 Hours)**

Stack:stack representation using array and Linked list. Applications of stack: Recursion, Validity of parentheses, Expression conversions and evaluations, mazing problem.

Queue: representation using array and Linked list, Types of queue, Applications of Queue:Job Scheduling etc.

SECTION-II Trees, Graphs, Hashing

Unit 4: Trees **(7 Hours)**

Trees: -Basic terminology, representation using array and linked list, Tree Traversals: Recursive and Non-recursive, Operations on binary tree: Finding Height, Leaf nodes, counting no of Nodes etc., Construction of binary tree from traversals, Binary Search trees (BST): Insertion, deletion of a node from BST. Threaded Binary tree (TBT): Creation and traversals on TBT, AVL tree.

Advanced Trees: Splay Tree, RB Tree, B,B+ Tree

Heap Tree (Min-Heap and Max-Heap)

Unit 5: Graphs **(7 Hours)**

Graphs:- Introduction to Graphs, Terminology (Vertices, Edges, Degree, Path, Cycle), Types of Graphs (Directed, Undirected, Weighted, Unweighted, Connected, Complete), Graph Representations (Adjacency Matrix, Adjacency List), Graph Traversal Techniques, Breadth-First Search (BFS), Depth-First Search (DFS), Spanning Trees, Minimum Spanning Tree (Prim's Algorithm, Kruskal's Algorithm), Shortest Path Algorithms (Dijkstra's Algorithm, Floyd–Warshall Algorithm), Topological Sorting, Applications of Graphs.

Unit 6: Hashing **(7 Hours)**

Hashing: -Introduction to Hashing, Need for Hashing, Applications of Hashing, Hash Tables, Hash Functions, Characteristics of a Good Hash Function, Division Method, Mid-square Method, Folding Method, Digit Extraction Method, Collision, Collision Resolution Techniques, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Load Factor, Rehashing, Insertion, Searching, Deletion, Performance Analysis, Time Complexity of Hashing Operations.

List of Practical's: (Any Eight)

1. Sorting+Searching
    1. Insertion/Bubble sort
    2. Merge Sort/Quick Sort
    3. Binary/Fibonacci Search

    Menu
    1. Read Array
    2. Insertion/Bubble Sort
    3. Merge Sort/Quick Sort
    4. Search

    Exit

2. Hashing
    1.Implement Linear probing in Hashing - Mod 11 slot 2/3/4
    2. Implement Program for Cuckoo Hashing

3. Stack -Implementation of Matching Parenthesis problem
    1. Conversion of Infix to Postfix
    2. Conversion of Postfix to Infix
    3. Evaluation of Postfix

4. Queue
    1. Implementation of Circular Queue
    2. Reversing Stack using Queue

5. Linked List
    A.Implementation of Singly Linked List
        1. Insert from Front, End, at given position
        2. Delete from Front, End, at given position
        3. Display list
    B. Implement Doubly linked list to
        1. Insert node
        2. Find nodes having even numbers
        3. Swap first and last node

8. Tree
    1. Recursive

a. Inorder Tree Traversal
b. Preorder Tree Traversal
c. Postorder Tree Traversal
d. Various Tree functions like mirror image,count

2. Non Recursive traversal for all traversal

9. Implement AVL Tree

10. Graph/Hashing
Implement BFS & DFS algorithm- Graph

11. Implement shortest path using Dijkstra's algorithm

12. AI-based assignment to 'generate test cases for data structure implementations/' generate visualization of advanced trees'/ 'implement decision
tree for various applications.

List of Course Projects(Any 1):
1. Problem solving using stack(like Tower of Hanoi)
2. Expression conversion like infix to prefix and postfix and vice versa
3. Problem solving using stack(like Job Scheduling)
4. String processing-Dictionary and Search enginesusing file and tree
5. Josephus problem
6. Site map using tree
7. Space partitioning using tree
8. Site Map using Graph
9. Database management using tree
10. Database management system using file structures and Hashing techniques
11. Travelling salesman problem
12. N queen problem
13. Hospital patient queue management system (priority queue, linked list).
14. Maze solver using stack,queue,and recursion.
15. Maze solver using graph.
16. Railway Reservation Queue Simulation.
17. Phone directory using BST.

Text Books: (As per IEEE format)
1. Fundamentals of Data Structures in C", E. Horwitz , S. Sahani, Anderson-Freed, Second Edition, Universities Press.
2. Data structures using C and C++", Y. Langsam, M.J. Augenstein, A.M.Tenenbaum, Pearson Education, Second Edition
3. "Data Structures and Algorithm Analysis in C++", Mark Allen Weiss, Pearson Education, Fourth Edition.

Reference Books: (As per IEEE format)

1. An Introduction to data Structures with applications", J. Tremblay, P. soresan, TMH Publication, 2nd Edition.2.
2. "Data Structures and Algorithms Made Easy", Narasimha Karumanchi, CareerMonkPublications,Fifth edition.
3. "Data Structures Using C", ReemaThareja, Oxford University Press, Second Edition.

Course Outcomes:

The student will be able to –

1. To interpret and diagnose the properties of data structures with their memory representations and time complexity analysis.(2)
2. To use linear data structures like stacks, queues etc. with their applications(3)
3. To handle operations on various data structures with the help of dynamic storage representation.(4)
4. To demonstrate Non-linear data structures like tree and perform various operations on it.(5)
5. To handle the operations on Graph data structure and to solve the applications ofGraph data structure.(4)
6. To design and analyze the appropriate data structure by applying various hashingTechniques.(2)

Moocs Links and additional reading material: (If any)

1. NPTEL – Programming, Data Structures and Algorithms Using Python

https://nptel.ac.in

2. NPTEL – Data Structures and Algorithms

https://nptel.ac.in

3.Coursera – Data Structures and Algorithms Specialization (University of California San Diego)

https://www.coursera.org

4.edX – Data Structures Fundamentals

https://www.edx.org

5.GeeksforGeeks – Data Structures Tutorial

https://www.geeksforgeeks.org/data-structures/

6.MIT OpenCourseWare – Introduction to Algorithms and Data Structures

https://ocw.mit.edu

Future Courses Mapping:

1. Advanced Algorithms

2. Graph Theory and Network Algorithms

3. Parallel and Distributed Algorithms

Job Mapping: (Optional)

1. Software Engineer / Software Developer

2. Data Scientist

3. Software Performance Engineer

CO - PO Mapping:

	Program Outcomes (PO)											PSO	
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO2
CO1	3	3	1	2	2	-	1	-	-	-	-	2	1
CO2	2	2	3	-	-	-	-	-	-	-	-	3	2
CO3	3	-	3	-	-	2	-	-	-	-	-	3	3
CO4	1	-	-	-	3	3	-	-	1	-	2	3	3
CO5	2	3	2	-	-	-	-	-	-	-	-	3	2
CO6	2	-	3	-	-	-	-	2	2	3	-	3	3
Average	2.17	2.67	2.40	2.00	2.5	2.5	1	2	1.5	3	2	2.83	2.33

FF No 654

AI 2302: Foundations of Intelligent Digital Systems

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures : 2 Hrs/week	CVV:20 Marks
Practical : 2 Hrs/week	Lab Assessment: 10 Marks
	ESE : 40 Marks

Course Prerequisites:

1. Basic electronics system

Course Objectives:

1. Represent and simplify logical functions using standard Boolean algebra and logic representation techniques.
2. Explain the principles and operation of digital logic circuits.
3. Design and analyze combinational and sequential logic circuits for engineering applications.
4. Describe the architecture, organization, and operation of microprocessor-based systems.
5. Apply microprocessor instruction sets and programming techniques to develop assembly language programs.
6. Analyze interrupt mechanisms and implement interrupt service routines in microprocessor-based systems.

Course Relevance:

This course is offered to Second Year B.Tech (Artificial Intelligence & Data Science) Engineering students and is designed to develop a strong foundation in Digital Electronics and Microprocessor Systems. The course enables students to analyze, design, and implement combinational and sequential digital circuits used in real-world applications.

SECTION-I

Unit 1: Digital Fundamentals **(4 Hours)**

Number Systems – Decimal, Binary, Octal, Hexadecimal, 1s and 2s Complements, Codes – Binary, BCD, Excess 3, Gray, Alphanumeric codes, Boolean Theorems, Logic Gates, Universal Gates, Sum of Products and Product of Sums, Minterms and Maxterms.

Unit 2: Combinational Digital Circuits **(5 Hours)**

Standard representation for Logic Functions, Simplification of logic functions using K-Map, Use of Don't Care Conditions. Code Converters: Binary, BCD Code, Gray Code, Excess-3 Code, 4-bit Binary Adder & Sub-tractor, Multiplexers & De-multiplexers, Encoder, Decoders, Parity Generator and Checker.

Unit 3: Sequential Circuit **(5 Hours)**

Introduction of flip-flop (F.F), 1-bit memory cell, clocked S-R, J-K, T, D Flip-flop: Truth Table, Excitation Table, Characteristics Table, Shift Register, Asynchronous and Synchronous Counters,
Sequence Generator, Sequences Detector (Moore and Mealy), **Application of AI in Digital System.**

SECTION-II

Unit 4: Introduction to 8086 Microprocessor **(4 Hours)**

Internal Architecture, Generation of Physical Address in 8086, 8086 Memory Segmentation, Register Organization, Addressing Modes: Immediate Addressing, Register Addressing, Direct Addressing, Indirect Addressing, Relative Addressing, Indexed Addressing, Bit Inherent Addressing, Bit Direct Addressing.

Unit 5: 8086 Instructions Types **(5 Hours)**

Instruction Types, Formats, Timings, Data Transfer Instructions, Arithmetic Instructions, Logical Instructions, Branch Instructions, Subroutine Instructions, Bit Manipulation Instruction, 8086 Pin Functions: Minimum & Maximum Mode System, Ready and Reset Pin Significance.

Unit 6: Interrupt Structure and Programmable Interface **(5 Hours)**

Interrupt Structure, Interrupt Service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR, NMI, Interrupt Response, Execution of an ISR, Priority of Interrupts, 8259 Programmable Interrupt Controller (PIC), Operating modes of 8259, Interfacing of 8259 with 8086 processor.

List of Practical's: (Any Six)

1. Verification of Logical Gates and Boolean Algebra (logic.ly/CEDAR AI Based Simulator).
2. Design Code converters e.g. Excess-3 to BCD and vice versa using logical gates.
3. Design Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Design Decoder – e.g. 2-bit comparator (IC 74138).
5. Design Synchronous Up /down counter using JK flip-flop.
6. Implementation of Sequences detector using JK flip flop.
7. Study of 8086 Architecture and Execution of sample programs.
8. Write 8086 ALP to find and count negative and positive number from signed array stored in memory and display magnitude of negative numbers (Workik.com /Context Driven AI-Assembler).
9. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and

display grade according to it.

10. Write 8086 ALP to perform block transfer operation. (Don't use string operations). Data bytes in a block stored in one array transfer to another array.
11. Write 8086 ALP for following operations on the string entered by the user.
    a) String length
    b) Reverse of string
    c) Palindrome
12. Write an ALP in 8086 to multiply two 8/16-bit numbers (Successive addition). Display and store the result.
13. Write an ALP in 8086 to multiply two 8/16-bit numbers (Add and Shift method). Display and store the result.
14. Write a 8086 ALP for file handling
    a) Create and Display File
    b) Copy File

List of Course Projects:

1. Design and Implement a Weather Imaging CubeSat with Telemetry Data Transmission.
2. Develop an Intelligent E-Bike Speed Controller System.
3. Design and Implement a Smart Wearable for Air and Water Pollution Monitoring.
4. Develop a Solar-Powered Marine Weather and Pollution Monitoring Buoy.
5. Design and Implement a Coin-Operated Water ATM with Automated Bottle Dispenser.
6. Implement an ARM-Based Multi-City Load Shedding Management System.
7. Design and Develop a Wireless Biomedical Parameter Monitoring System Using ARM9.
8. Implement an ARM and RFID-Based Smart Security System for Home, Office, and Industrial Applications.
9. Develop an Advanced ARM-Based Electronic Voting Machine (EVM) .
10. Design an Online Parallel Examination and Assessment System.
11. Design and Develop AI, Machine Learning, Deep Learning, and Blockchain-Based Solutions for Agriculture, Healthcare, Education, Government, Transportation, Banking, and Insurance Applications.

Text Books: (As per IEEE format)

1. *Douglas Hall, "Microprocessors and Interfacing",2nd Edition, Tata McGraw Hill Publications, ISBN 0-07-025742-6.*
2. *"Advanced 80386, programming techniques " , James Turley , Tata McGraw Hill Publications, ISBN – 0-07-881342-5*
3. *Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630- 011, December 1995.*
4. *R.P. Jain, "Modern Digital Electronics,"3rdEdition, Tata McGraw-Hill, 2003, ISBN 0 - 07 - 049492 – 4.*

Reference Books: (As per IEEE format)

1. *Ray Duncan, "Advanced MS DOS Programming," 2nd Edition BPB Publications ISBN 0 – 07 – 048677 – 8.*
2. *M. Mano, "Digital Design", 3rd Edition, Pearson Education, 2002, ISBN - 81 - 7808 – 555 – 0.*
3. *A. Malvino, D. Leach, "Digital Principles and Applications", 5th Edition, Tata McGraw Hill, 2003, ISBN 0 - 07 - 047258 – 05.*

Course Outcomes:

The student will be able to –

1. Illustrate the standard representation for logical functions (1).
2. Explore the knowledge based on combinational logic circuits (2).
3. Design applications based sequential circuits (2).
4. Demonstrate the concepts of microprocessor systems (3).
5. Classify and compare types of microprocessor instructions (3).
6. Illustrate the concept of interrupts and its service routine (4).

Moocs Links and additional reading material: (If any)

1. NPTEL – Digital Circuits
   - **https://nptel.ac.in/**
   - Topics: Boolean Algebra, Logic Gates, Combinational & Sequential Circuits
2. NPTEL – Microprocessors and Microcontrollers
   - **https://nptel.ac.in/**
   - Topics: 8086 Architecture, Instruction Set, Interrupts, Interfacing

3. Coursera – Digital Systems: From Logic Gates to Processors
    - **https://www.coursera.org/**
    - Covers digital logic, CPU architecture, and processor fundamentals.
4. edX – Embedded Systems and Digital Design
    - **https://www.edx.org/**
    - Focuses on digital system design and embedded computing.
5. MathWorks – MATLAB Academy
    - **https://matlabacademy.mathworks.com/**
    - Digital logic, simulation, and embedded system fundamentals.
6. Cisco Networking Academy – Introduction to IoT and Digital Transformation
    - **https://www.netacad.com/**
    - Covers IoT devices, embedded systems, and intelligent digital technologies.

Future Courses Mapping:

1. Deep Learning
2. Data Mining
3. Cloud Computing –AWS
4. IOT
5. Artificial Intelligence
6. Pattern Recognition
7. Natural Language Processing
8. Computer Vision

Job Mapping: (Optional)

1. Data Architect.
2. Infrastructure Architect
3. Enterprise Architect

CO - PO Mapping:

	Program Outcomes (PO)											PSO	
CO/ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	1	2		1			1			1	1	1
CO2	2	2	2		1	2					1	1	1
CO3	2		2		1	2					1		1
CO4	2		2		1	2		1			1		1
CO5	2	1	2			2		1			1	1	
CO6	2										1	1	
Average	2	1.33	2		1	2		1			1	1	1

FF No: 654

AI 2303: Database Management Systems

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	Lab Assessment: 10 Marks
Tutorial :	0 Hrs/week	ESE: 40 Marks

Course Prerequisites:

1. Data structures
2. Discrete Mathematics

Course Objectives:

1. To introduce the core principles and architectures of modern database systems.
2. To apply data modelling techniques using Entity-Relationship and relational models.
3. To design and normalize relational schemas for efficient data storage.
4. To write effective SQL and PL/SQL programs for data manipulation and transaction control.
5. To understand the concepts of query processing, optimization, and indexing.
6. To explore the role of NoSQL databases, Big Data systems, and distributed databases in modern applications.

Course Relevance: The course is offered in S.Y. B.Tech. AIDS Department.

SECTION-I

Unit 1: Introduction to Database Systems: Need of DBMS, Design of DBMS, characteristics, components of DBMS, DBMS vs File Systems, Applications of DBMS in AI/ML **(2 hours)**

Unit 2: Data Models and Relational Model: Types of data models, Relational model: schema, keys, integrity constraints**. (2 hour)**

Unit 3: Entity-Relationship (ER) Modeling: ER diagrams, attributes, relationships, Enhanced ER concepts: generalization, specialization, Codd's Rules. **(2 hours)**

Unit 4: SQL Basics: Set, string, Date and numerical function, selection, projection, SQL: DDL, DML, DCL, TCL, Aggregate Function, Group by and having clause. **(4 hours)**

Unit 5: PL-SQL: Subqueries, Procedure, Function, Trigger, Cursors, joins, views, Nested

Queries. **(4 hours)**

SECTION-II

Unit 6: Normalization and Schema Design: Closure, Minimal Cover, Functional dependencies, Multivalued Dependency, Decomposition; Lossless Join and Dependency Preservation; 1NF, 2NF, 3NF, BCNF,4NF. **(4 hours)**

Unit 7: Transaction Management & Concurrency Control: Basic Concept, ACID properties, State diagram, Concept of Schedule, Serial Schedule, Serializability-Conflict and View, Concurrency control Protocols. **(4 hours)**

Unit 8: NoSQL Databases: Introduction to NoSQL and comparison with RDBMS, **MongoDB Essentials:** Collections vs tables, Documents and BSON format, CRUD operations in MongoDB, Aggregation functions**. (3 hours)**

Unit 9: Emerging database Technologies: Parallel Databases, Internet Databases, Mobile databases, Cloud Databases, SQLite Database, I/O Parallelism, Distributed Database Systems. **(3 hours)**

List of Practical's:

1) Choose a database application; you propose to work on throughout the course. Perform requirement analysis in detail for the same. Draw an entity-relationship diagram for the proposed database.

2) Create a database with appropriate constraints using DDL and populate/modify it with the help of DML.

3) Design and Execute "SELECT" queries using conditional, logical, like/not like, in/not in, between...and, is null/is not null operators in where clause, order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.

4) Write equijoin, non-equijoin, self-join and outer join queries. Write queries containing single row / multiple row / correlated sub queries using operators like =, in, any, all, exists etc. Write DML queries containing sub queries. Study a set of query processing strategies.

5) Write PL/SQL blocks to implement all types of cursors.

6) Write useful stored procedures and functions in PL/SQL to perform complex computation.

7) Write and execute all types of database triggers in PL/SQL.

8)Execute DDL statements which demonstrate the use of views. Try to update the base table

using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.

9) Create a database with suitable example using MongoDB and implement Inserting and saving document, Removing document, Updating document

10) Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques: find and findOne, Query criteria, Type-specific queries

11) Implement Map Reduce operation with suitable example using MongoDB.

12) Write a Python script that connects to a **MongoDB database** storing **User Data**.

List of Course Projects:

1. E-Commerce Order Management System
2. Student Academic Performance Management System
3. Online Movie Recommendation System
4. Personal Finance Management System
5. Real-Time Inventory Management System
6. Online Bookstore with Reviews and Recommendations
7. Health Monitoring System Database
8. Smart Traffic Management System
9. Social Media Data Analysis and Insights
10. Travel and Booking System
11. AI-Based Personalized Health Recommendation System
12. AI-Powered Fraud Detection System for Transactions
13. Intelligent Chatbot for Customer Support
14. AI-Based Predictive Maintenance System for Manufacturing
15. AI-Driven Job Recommendation System
16. AI-Powered Sentiment Analysis for Customer Reviews
17. AI-Enhanced Inventory Forecasting System
18. AI-Based Smart Home Automation System
19. Predictive Analytics for Real Estate Investment
20. AI-Driven Disease Diagnosis System

Text Books: (As per IEEE format)

1. *Abraham Silberschatz, Henry F. Korth, S. Sudarshan; "Database System Concepts"; 6th Edition, McGraw-Hill Education*
2. *Ramez Elmasri, Shamkant B. Navathe; "Fundamentals of Database Systems"; 7th Edition, Pearson*

Reference Books: (As per IEEE format)

1. Thomas M. Connolly, Carolyn E. Begg," Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition ;Pearson

2. Raghu Ramakrishnan, Johannes Gehrke; "Database Management Systems", 3rd Edition; McGraw Hill Education

3. Kristina Chodorow, MongoDB The definitive guide, O'Reilly Publications, ISBN: 978-93-5110- 269-4, 2nd Edition.

4. Dr. P. S. Deshpande, SQL and PL/SQL for Oracle 10g Black Book, DreamTech.

5. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publication.

6. Reese G., Yarger R., King T., Williums H, Managing and Using MySQL, Shroff Publishers and Distributors Pvt. Ltd., ISBN: 81 - 7366 - 465 – X, 2nd Edition.

7. Dalton Patrik, SQL Server – Black Book, DreamTech Press.

8. Eric Redmond, Jim Wilson, Seven databases in seven weeks, SPD, ISBN: 978-93-5023-918-6.

9. Jay Kreibich, Using SQLite, SPD, ISBN: 978-93-5110-934-1, 1st edition.

Moocs Links and additional reading material:

1. https://nptel.ac.in/courses/106/105/106105175/
2. https://onlinecourses.nptel.ac.in/noc21_cs04/preview
3. https://www.datacamp.com/courses/introduction-to-sql Oracle MOOC: PL/SQL Fundamentals - Oracle APEX

Course Outcomes

1. Design and construct conceptual database models using ER and EER diagrams for real- life applications.

2. Transform high-level data models into normalized relational schemas using functional dependencies and synthesis techniques.

3. Apply the concepts of normalization to develop the quality relational data model

4. Formulate and execute queries using relational algebra, SQL, and develop procedural constructs using PL/SQL.

5. Explore and implement modern database technologies such as NoSQL and Big Data frameworks like MongoDB and Hadoop.

6. Demonstrate understanding of physical database structures, indexing mechanisms, and query optimization techniques

Future Courses Mapping:

1. Introduction to Databases and AI Integration
2. Relational Data Models and Normalization
3. Advanced Database Development with PL/SQL
4. Indexing, Query Optimization, and AI
5. AI-Driven Data Management Systems

Job Mapping:

Job opportunities that one can get after learning this course

1. Database Administrator (DBA)
2. Data Scientist
3. Data Engineer
4. Machine Learning Engineer
5. Business Intelligence (BI) Analyst
6. AI/ML Data Analyst
7. Cybersecurity Analyst (SQL Injection Focus)
8. Cloud Database Architect
9. Full Stack Developer (with AI/ML focus)
10. AI/ML Consultant (Database Solutions.

	Program Outcomes (PO)											PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	3	3	1	1	0	0	0	1	1	1	1	2
CO2	2	3	3	1	1	0	0	0	0	0	0	1	2
CO3	2	3	3	2	2	0	0	1	0	0	1	1	2
CO4	1	2	2	1	1	0	0	0	0	0	0	1	2
CO5	2	2	1	2	1	1	0	0	1	1	1	1	2
CO6	2	2	2	1	2	1	0	1	1	1	0	1	2
Average	1.83	2.50	2.33	1.33	1.33	0.33	0.00	0.33	0.50	0.50	0.50	1.00	2.00

FF No:654

AI 2304 : Discrete Structures and Computational Theory

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	MSE (W):40 Marks
Tutorial :	0 Hrs/week	Lab Assessment: 10Marks

Course Prerequisites:

1. Mathematics

Course Objectives:

1. To use appropriate set, function and relation models to understand practical examples, and interpret the associated operations and terminologies in context.
2. Determine number of logical possibilities of events.
3. To introduce the students to basics of Theory of Computation
4. To study abstract computing models to provide a formal connection between algorithmic problem solving and the theory of languages
5. To learn Grammar, Pushdown Automata for language processing and algorithm design

Course Relevance:

The course is offered in S.Y. B.Tech. to all branches of Engineering

SECTION-I

Unit I Set Theory and Logic **5** Hours

Introduction and significance of Discrete Mathematics, Sets– Naïve Set Theory (Cantorian Set Theory), Axiomatic Set Theory, Set Operations, Cardinality of set, Principle of inclusion and exclusion. Types of Sets – Bounded and Unbounded Sets, Diagonalization Argument, Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets, Power set, Propositional Logic- logic, Propositional Equivalences, Proof by Mathematical Induction and Strong Mathematical Induction

#Exemplar/Case Studies Know about the great philosophers- Georg Cantor, Richard Dedekind and Aristotle

Mapping of Course Outcomes for Unit I CO1

Unit II Relations and Functions **4** Hours

Relations and their Properties, n-ary relations and their applications, Representing relations , Closures of relations, Equivalence relations, Partial orderings, Partitions, Hasse diagram, Lattices, Chains and Anti-Chains, Transitive closure. Functions- Surjective, Injective and Bijective functions, Identity function, Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions.

#Exemplar/Case Studies Know about the great philosophers-Dirichlet

Mapping of Course Outcomes for Unit II CO2

Unit III Formal Language Theory and Finite Automata **5** Hours

Finite Automata (FA): An informal picture of FA, Finite State Machine (FSM), Language accepted by FA, Definition of Regular Language. FA without output: Deterministic and Nondeterministic FA (DFA and NFA), epsilon- NFA and inter-conversion. Minimization of DFAs. FA with output: Moore and Mealy machines -Definition, models, inter-conversion.

#Exemplar/Case Studies FSM for vending machine, spell checker

*Mapping of Course Outcomes for Unit III CO3

SECTION-II

Unit IV Regular Expressions (RE) **4** Hours

Introduction, Operators of RE, Precedence of operators, Algebraic laws for RE, Language to Regular Expressions, Equivalence of two REs. Conversions: RE to NFA, DFA, DFA to RE using Arden's theorem, Pumping Lemma for Regular languages, Closure and Decision properties of Regular languages. Myhill-Nerode theorem.

#Exemplar/Case Studies RE in text search and replace

*Mapping of Course Outcomes for Unit IV CO4

Unit V Context Free Grammar (CFG) **5** Hours

Basic Elements of Grammar, Formal Definition of Context Free Grammar, Sentential form, Derivation and Derivation Tree/ Parse Tree, Context Free Language (CFL), Ambiguous Grammar, writing grammar for language. Simplification of CFG: Eliminating Є-productions, unit productions, useless production, useless symbols. Normal Forms: Chomsky Normal Form, Greibach Normal Form, Pumping Lemma for CFG, Closure properties of CFL, Decision properties of CFL, Chomsky Hierarchy, Cock-Younger-Kasami Algorithm.

#Exemplar/Case Studies Parser, CFG for Palindromes, Parenthesis Match

*Mapping of Course Outcomes for Unit V CO5

Unit VI Pushdown Automata (PDA) & Turing Machine **5** Hours

Introduction, Formal definition of PDA, Equivalence of Acceptance by Final State and Empty stack, Non-deterministic PDA (NPDA), PDA and Context Free Language, Equivalence of PDA and CFG, PDA vs CFLs. Deterministic CFLs. Turing Machine Model, Formal definition of Turing Machines, Language Acceptability by Turing Machines, Design of TM

#Exemplar/Case Studies Parsing and PDA: Top-Down Parsing, Bottom-up Parsing simulation showing use of PDA

*Mapping of Course Outcomes for Unit VI CO6

List of Practical's: (Any Eight)

Students are encouraged to use Python programming for the practical assignments.

1. Write a program to perform the following operations on two user-defined sets.

   1. Union 2. Intersection 3. Difference 4. Symmetric Difference

   5. Power Set 6. Cardinality

2. To solve real-life problems using the Principle of Inclusion and Exclusion.

   - 60 students like Cricket
   - 40 like Football
   - 25 like both

   Find

   - Students liking at least one game
   - Students liking only Cricket
   - Students liking only Football

3. To construct truth tables and verify logical equivalence for

   1. $P \rightarrow Q$
   2. $P \leftrightarrow Q$
   3. $(P \vee Q) \wedge \neg P$
   4. $(P \rightarrow Q) \wedge (Q \rightarrow R)$

4. To identify different types of functions. Determine whether each function is

   - Injective
   - Surjective
   - Bijective

   Find

   1. Inverse Function 2. Composition of Functions

5. Develop a program to simulate Deterministic Finite Automata (DFA) for string acceptance. Understand DFA implementation and state transitions.
6. Develop a program to simulate Non-Deterministic Finite Automata (NFA). Understand non-deterministic computation.
7. Implement NFA to DFA conversion using subset construction algorithm. Learn automata conversion techniques.
8. Design a program to simulate Moore and Mealy Machines and compare their outputs. Learn finite state transducers.
9. Develop a program to convert Context-Free Grammar (CFG) to Chomsky Normal Form (CNF). Learn grammar normalization.
10. Develop a program to compute FIRST and FOLLOW sets of a Context-Free Grammar. Understand grammar analysis

List of Course Projects:

1. Set Theory Calculator
2. Logic Expression Evaluator
3. Relation Analyzer
4. Hasse Diagram Generator
5. Spell Checker using Finite Automata
6. DFA Minimization Tool
7. Regular Expression Validator
8. DFA to Regular Expression Converter
9. Text Search Utility using Regular Expressions
10. Parse Tree Generator
11. Parenthesis Matching using PDA
12. Turing Machine Simulator

Text Books: (As per IEEE format)

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman,"Introduction to Automata Theory Languages and Computation", Addison-Wesley, ISBN 0-201-44124-1
2. John Martin, "Introduction to Languages and The Theory of Computation", 2nd Edition, McGrawHill Education, ISBN-13: 978-1-25-900558-9, ISBN-10: 1-25-900558-5

Reference Books: (As per IEEE format)

1. Bernard Kolman, Robert C. Busby and Sharon Ross, ―Discrete Mathematical Structures‖, Prentice-Hall of India /Pearson, ISBN: 0132078457, 9780132078450.
2. Eric Gossett, "Discrete Mathematical Structures with Proofs", Wiley India Ltd, ISBN:978-81 265-2758-8.
3. Daniel Cohen, "Introduction to Computer Theory", Wiley & Sons, ISBN 97881265133454.
4. J. Carroll & D Long, "Theory of Finite Automata", Prentice Hall, ISBN 0-13-913708-45.
5. Kavi Mahesh, "Theory of Computation: A Problem-Solving Approach", Wiley India, ISBN1081265331106.

Moocs Links and additional reading material:

1. https://cglab.ca/~michiel/TheoryOfComputation/TheoryOfComputation.pdf
2. https://www.cs.virginia.edu/~robins/Sipser_2006_Second_Edition_Problems.pdf
3. http://ce.sharif.edu/courses/94-95/1/ce414-

Course Outcomes:

Upon completion of the course, student will be able to –

CO1: Design and analyze real world engineering problems by applying set theory, propositional logic and mathematical induction

CO2: Develop skill in expressing mathematical properties of relation and function

CO3: Understand formal language, translation logic, essentials of translation, alphabets, language representation and apply it to design Finite Automata and its variants

CO4: Construct regular expression to present regular language and understand pumping lemma for RE

CO5: Design Context Free Grammars and learn to simplify the grammar

CO6: Construct Pushdown Automaton model for the Context Free Language

Future Courses Mapping:

1. Compiler
2. System Programming

Job Mapping:

Job opportunities that one can get after learning this course

1. Research Scientist
2. Compiler Designer / Software Engineer

CO - PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	2	1	-	-	1	-	2	3	-
CO2	3	2	2	2	1	-	-	-	-	-	1	2	-
CO3	3	3	3	2	3	-	-	-	-	-	1	3	2
CO4	3	3	3	2	3	-	-	-	-	-	1	3	2
CO5	2	2	3	1	2	-	-	-	-	-	1	2	2
CO6	2	2	3	2	2	-	-	-	-	-	1	3	2

FF No:654

MM0105: Augmented Reality and Virtual Reality

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	0 Hrs/week	HA : 20 Marks
Tutorial :	1 Hrs/week	MSE (W) : 40Marks

Course Objectives:

1. To Understand immersive technologies, their taxonomy, tools, applications, and distinguish AR, VR, and MR.
2. To Learn core concepts, hardware, and development lifecycle of Virtual Reality systems.
3. To gain practical knowledge of VR application development using tools like Unity, Unreal Engine, and Blender, including basic scripting and templates.
4. To comprehend Augmented Reality principles, types, and image processing techniques
5. To familiarize with AR development tools such as Vuforia and Unity, addressing challenges and real-world AR application deployment
6. To explore Mixed Reality technologies, their use cases, security challenges, and technical requirements for MR application development

Course Relevance: The course is offered in S.Y. B.Tech. AIDS Department.

SECTION-I

Unit 1: Introduction - (5 Hours)

Immersive Basics, taxonomy, methods and techniques, tools, introduction to Virtual Reality, Augmented Reality and Mixed Reality, spectrum, difference between AR/VR/MR, Applications of AR/VR/MR, Online and Offline tools for AR/VR/MR, Consequences of Excessive AR/VR/MR.

Unit 2: Virtual Reality - (5 Hours)

Basics of Virtual Reality, Types, Augmented Reality, Head Mounted Device (HMD), Types and working along with different applications, Collaborative VR, Key components and benefits, VR Application Development Cycle, Comparison with Software Development Life Cycle, Web VR, Mobile VR. Tools, Use cases, Evolution of VR technology, Tools and Technologies for VR app development.

Unit 3: Virtual Reality App Development - (4 Hours)

Introduction to Virtual Reality App development, Software's, Unity, Blender, Unreal Engine, Blueprint, Types, Templates in Unreal Engine development, Basics of Unity and, introduction to game development using unity.

SECTION-II

Unit 4: Augmented Reality - (5 Hours)

Introduction to Augmented Reality, types, mobile AR/ Web AR-headset AR, working principle, image processing principles in AR app, Basics of image processing, feature points detection, Keypoint detection, Keypoint description.

Unit 5: Augmented Reality App Development - (5 Hours)

Tools and technologies for AR app development, Vuforia, Vuforia engine, functions, Vuforia and Unity, challenges of AR app development, Applications of Augmented Reality in real world scenarios.

Unit 6: Mixed Reality - (4 Hours)

Mixed Reality basics, Technological requirement in development and deployment of MR apps, Security issues with MR apps. Mixed Reality applications. Use case of mixed reality.

List of Tutorials:

Tutorial 1: To study and prepare a report on **Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR)** through a hands-on demonstration using the **ARLOOPA** application.
Tutorial 2: To create a **Unity** project in which a 3D cube moves using the **Arrow Keys** or **WASD** keyboard controls.

Tutorial 3: To implement a **Camera Follow** system in Unity that enables the camera to smoothly track the movement of a 3D player object.

Tutorial 4: To design and develop a basic **Virtual Reality (VR)** scene using the **A-Frame** framework, demonstrating fundamental VR concepts and interactions.

Tutorial 5: To create a **VR-like interactive environment** in Unity where the player can navigate using the keyboard and mouse and perform object interactions such as picking up and dropping objects without requiring a VR headset.

Tutorial 6: To create a **bouncing ball animation** in **Blender** using keyframes and the timeline, demonstrating fundamental animation principles such as **timing, squash and stretch, anticipation, and easing** to simulate realistic motion.

Tutorial 7: To develop an interactive AR application in Unity using Vuforia Engine that recognizes an image target and displays a 3D object on it, demonstrating the fundamentals of marker-based Augmented Reality.

Tutorial 8: To create a simple first-person 3D game in Unity by designing a basic environment with obstacles, implementing collision detection, score collection, and a game-over condition using C# scripting.

List of Course Projects:

1. Virtual Campus Tour
2. AR-Based Furniture Placement
3. 3D Maze Game
4. VR Escape Room
5. Virtual Museum
6. Virtual Art Gallery
7. AR Solar System
8. AR Animal Learning App
9. Virtual Classroom
10. VR House Walkthrough
11. First-Person Adventure Game
12. 12 Endless Runner Game
13. Treasure Hunt Game
14. Car Parking Simulator
15. Traffic Signal Simulation
16. Virtual Shopping Mall
17. AR Business Card
18. Interactive Science Laboratory
19. Virtual Zoo
20. 3D Puzzle Game

List of Course Seminar Topics:

1. To study the fundamentals of Augmented Reality (AR) and its real-world applications.
2. To explore the concepts, components, and applications of Virtual Reality (VR).
3. To understand the principles of Mixed Reality (MR) and its role in immersive computing.
4. To study Extended Reality (XR) and its impact on the future of digital interaction.
5. To explore the architecture, features, and applications of the Metaverse.
6. To study the use of Unity Engine for developing interactive AR and VR applications.
7. To understand the role of Blender in 3D modeling, animation, and game asset creation.
8. To explore the features and applications of the A-Frame framework for WebVR development.
9. To study the working principles of marker-based and markerless Augmented Reality systems.
10. To understand the applications of AR and VR technologies in the healthcare sector.
11. To study the use of AR and VR in education and immersive learning environments.
12. To explore the applications of Virtual Reality in the gaming and entertainment industry.
13. To understand the role of Digital Twin technology in smart industries and manufacturing.
14. To study the applications of Spatial Computing in next-generation computing systems.
15. To explore Human-Computer Interaction (HCI) techniques in AR and VR environments.
16. To understand gesture recognition and motion tracking technologies used in immersive systems.
17. To study the security, privacy, and ethical challenges associated with AR and VR technologies.
18. To explore the integration of Artificial Intelligence (AI) with AR and VR applications.
19. To study the latest trends and future developments in AR, VR, MR, and XR technologies.
20. To explore career opportunities and emerging research areas in AR, VR, XR, and game development.

Text Books: (As per IEEE format)

1. ***Tony Parisi , Learning Virtual Reality:*** *Developing Immersive Experiences and Applications for Desktop, Web, and Mobile, Wiley, 2015.*

2. ***Murray Ramirez, Virtual Reality for Beginners!:*** *How to Understand, Use & Create with VR, by, 2016.*

3. ***Roger Froze, Augmented Reality For Beginners!:*** *Principles & Practices for Augmented Reality & Virtual Computers, 2016*

Reference Books and E-Resources

1. *Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, 3D User Interfaces, Theory and Practice, Addison Wesley, USA, 2005.*
2. *Oliver Bimber and Ramesh Raskar, Spatial Augmented Reality: Merging Real and Virtual Worlds, 2005. Example - H. Schmidt-Walter, R. Kories; 'Electrical Engineering. A Pocket Reference'; Artech House, 2007. Accessed: Oct. 16, 2016. [Online]. Available: https://ebookcentral.proquest.com*

Course Outcomes:

1. Describe the key concepts and applications AR/VR/MR.
2. Apply the AR/VR application development life cycle using different tools.
3. Develop VR applications using suitable VR tools.

4. Understand Augmented Reality and image processing.
5. Develop AR applications using suitable AR tools.
6. Understand advances in Mixed Reality.

For MOOCs and other learning Resource

1. https://www.deepar.ai/demos
2. https://app.vectary.com/
3. https://builtin.com/articles/what-is-webar

Future Courses Mapping:

1. Provides the basic knowledge of AR, VR, and MR technologies.
2. Helps students develop AR and VR applications using modern tools.
3. Builds a foundation for game development and 3D visualization.
4. Prepares students for advanced courses in Extended Reality (XR) and the Metaverse.
5. Enhances practical skills in Unity, Blender, and AR development.
6. Supports career opportunities in AR/VR development, simulation, and immersive technologies.

Job Mapping:

This course equips students with the knowledge and practical skills required for careers in AR/VR, Unity, and immersive application development. It prepares students for entry-level roles such as AR Developer, VR Developer, Unity Developer, and Game Developer.

CO - PO Mapping:

	Program Outcomes (PO)											PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	1										1		
CO2			2										
CO3			3		2	1						2	3
CO4	1												
CO5			3		2	1						2	3
CO6								2					
Average	1		2.6		2	1		2			1	2	3

FF No:654

MM0106: Image Processing

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	0 Hrs/week	HA : 20 Marks
Tutorial :	1 Hrs/week	MSE (W) : 40Marks

Course Prerequisites:

1. Basic knowledge of computer programming (Python/MATLAB/C preferred).
2. Fundamental understanding of mathematics, including matrices and vectors.
3. Basic concepts of computer graphics and digital images.

Course Objectives:

1. To understand the basics of digital images, color models, and image formats.
2. To apply enhancement techniques in spatial and frequency domains.
3. To learn segmentation and feature extraction methods.
4. To study image compression techniques and their applications.
5. To use image transforms for image analysis and processing.
6. To apply image processing methods to real-time problem solving.

Course Relevance:

Provides the fundamentals of digital image processing for applications in AI, Computer Vision, Medical Imaging, and Robotics.

SECTION-I

Unit 1: Introduction to Image processing **(5 Hours)**

Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YC_bC_r and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF,png, raster image format). Image sampling and quantization.

Unit 2: Image Enhancement **(5 Hours)**

Thresholding (Global, Adaptive, and Otsu's Method), Spatial Domain Techniques including Image Negative, Contrast Stretching, Histogram Processing and Histogram Equalization, Image Subtraction,

Image Smoothing using Mean, Gaussian, and Median Filters, Image Sharpening using High-Pass, Laplacian, and Sobel Filters, and Frequency Domain Techniques including Butterworth Low-Pass Filter and High-Pass Filter.

Unit 3: Image Analysis (5 Hours)

Image Segmentation: Threshold-based, Edge-based, Region-based, Watershed, and Clustering-based Segmentation; Edge Detection using Sobel, Prewitt, Canny, and Laplacian Operators. Feature Extraction: Boundary Representation (Chain Code, Fourier Descriptors), Region Properties (Area, Perimeter, Euler Number, Eccentricity, Moments), Texture and Color Features, and Introduction to CNN-based Feature Extraction and Image Segmentation.

SECTION-II

Unit 4: Image Compression (4 Hours)

Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), scalar and vector quantization

Unit 5: Object recognition (4 Hours)

Need of Object Recognition, Automated Object Recognition System, Pattern and Pattern Class, Relationship between Image Processing and Object Recognition, Approaches to Object Recognition: Template Matching, Feature-Based Recognitio, Haar Cascade Classifier.

Unit 6: Image Transform (5 Hours)

Introduction to two dimensional orthogonal and unitary transforms, properties of nitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine,Discrete Wavelet Transform (DWT),Walsh–Hadamard Transform (WHT) , Applications of transforms in Image processing.

List of Practical's: (Any ….)

List of Experiments

1. Write matlab code to display following binary images
Square, Triangle , Circle

- Write matlab code to perform following operations on images
- Flip Image along horizontal and vertical direction.
- Enhance quality of a given image by changing brightness of image.
- Image negation operation.
- Change contrast of a given Image.

2. Write Matlab code to implement pseudo colouring operation of a given image. Write Matlab Code for Pseudo Colour of Image by using Gray to colour transform.

3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.

4. Write matlab code to find following statistical properties of an image.

- Mean
- Median
- Variance
- Standard deviation
- Covariance.

5. Write matlab code to enhance image quality by using following techniques

- Logarithmic transformation
- Histogram Equalization
- Gray level slicing with and without background.
- Inverse transformation.

6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.

7. Write matlab code to enhance image quality by using following techniques

- Low pass and weighted low pass filter.
- Median filter.
- Laplacian mask.

8. Write matlab code for edge detection using Sobel, Prewitt and Roberts operators.

9. Write C-language code to find out Huffman code for the following word COMMITTEE.

10. Write matlab code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3,Y- 0.1).

11. Write matlab code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.

12. Develop an AI-inspired image stylization and enhancement system using classical image processing techniques.(GenAI based)

List of Tutorials:

1) Introduction to image processing tools and environment setup using Python or MATLAB.
2) Reading, displaying, and basic manipulation of images including color space conversions.
3) Image sampling and quantization with analysis of resolution and bit depth effects.
4) Spatial domain enhancement techniques like contrast stretching and histogram equalization.
5) Fourier Transform and frequency domain filtering for image analysis.
6) Noise addition and image restoration using mean, median, and inverse filtering.

7) Edge detection and morphological operations such as dilation and erosion.
8) Image compression techniques and color-based segmentation using the HSI model.

List of Course Projects:

1. Real-Time Multi-Object Detection and Tracking using YOLO and OpenCV
2. Automatic Face Recognition with Face Mask Detection using Deep Learning
3. Vision-Based Driver Drowsiness and Distraction Detection System
4. Deep Learning-Based Medical Image Segmentation for Tumor Detection
5. AI-Based Automatic Number Plate Recognition (ANPR) System
6. Image Forgery and Deepfake Detection using CNN and Image Forensics
7. Plant Leaf Disease Detection and Severity Analysis using Deep Learning
8. Crack Detection in Civil Infrastructure using CNN and Image Processing
9. Satellite Image Land Cover Classification using Deep Learning
10. Human Pose Estimation and Activity Recognition using Computer Vision
11. OCR-Based Intelligent Document Analysis and Information Extraction
12. Vision-Based Smart Attendance System using Face Recognition and Liveness Detection

List of Course Seminar Topics: (*if applicable to your course*)

1. Human Visual System (HVS) and Color Models in Digital Image Processing
2. Image Sampling and Quantization: Concepts and Applications
3. Image Enhancement Techniques in Spatial and Frequency Domains
4. Histogram Equalization and Contrast Enhancement Methods
5. Image Filtering Techniques for Noise Removal
6. Edge Detection Techniques: Sobel, Prewitt, Canny, and Laplacian
7. Image Segmentation Techniques and Their Applications
8. Feature Extraction Methods in Digital Images
9. Image Compression Standards: JPEG and Lossless Compression Techniques
10. Haar Features and Haar Cascade Classifier for Object Detection
11. Object Recognition Approaches: Template Matching and Feature-Based Recognition
12. Two-Dimensional Discrete Fourier Transform (DFT) in Image Processing
13. Discrete Wavelet Transform (DWT) for Image Analysis and Compression
14. Applications of Image Processing in Medical Imaging, Remote Sensing, and Industrial Inspection
15. AI-Assisted Image Processing: Integration of Classical Image Processing with Deep Learning

Text Books: (As per IEEE format)

1. Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition,2010.
2. Anil K. Jain, 'Fundamentals of Digital Image Processing', Pearson, 2002

Reference Books: (As per IEEE format)

1. Kenneth R. Castleman, 'Digital Image Processing', Pearson, 2006.
2. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, 'Digital Image Processing using MATLAB', Pearson
Education, Inc., 2011.
3. D,E. Dudgeon and RM. Mersereau, 'Multidimensional Digital Signal Processing', Prentice Hall Professional
Technical Reference, 1990
4. William K. Pratt, 'Digital Image Processing', John Wiley, New York, 2002
5. Milan Sonka et al 'Image processing, analysis and machine vision', Brookes/Cole, Vikas Publishing House, 2nd
edition, 1999.

Course Outcomes:

The student will be able to –

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

1. Describe various image models and their properties.
2. Apply spatial filtering techniques for image enhancement.
3. Identify and implement image segmentation techniques.
4. Apply appropriate lossless and lossy compression methods.
5. Implement basic object recognition techniques.
6. Use image transforms to analyze and process digital images.

Future Courses Mapping:

Computer Vision, Deep Learning, Pattern Recognition, and Artificial Intelligence.

Job Mapping: (Optional)

1. Image Processing Engineer
2. Computer Vision Engineer
3. AI/ML Engineer
4. Data Scientist
5. Robotics Engineer

6. Machine Vision Engineer

CO - PO Mapping:

	ProgramOutcomes(PO)PSO												
CO/PO	PO1	PO 2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	1	1	2	-	1	-	-	-	-	-	-	3	1
CO2	2	1	2	2	2	-	-	-	-	-	-	3	2
CO3	3	2	2	2	2	-	-	-	-	-	-	3	2
CO4	3	2	2	2	2	-	-	-	-	-	-	2	3
CO5	2	2	2	2	2	-	-	-	-	-	-	2	3
CO6	3	3	2	2	2	-	-	-	-	-	-	3	2
Average	**2.33**	**1.83**	**2**	**1.67**	**1.83**							**2.67**	**2.167**

FF No:654

MM0107: Computer Graphics

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	0 Hrs/week	HA : 20 Marks
Tutorial :	1 Hrs/week	MSE (W) : 40Marks

Course Objectives:

1. To understand the fundamentals and applications of computer graphics.
2. To study graphics hardware and the graphics pipeline.
3. To implement algorithms for drawing and transforming 2D/3D objects.
4. To explore concepts of projections, shading, and visibility.
5. To introduce animation techniques and graphics programming tools.
6. Describe the fundamentals of animations and gaming.

Course Relevance: The course is offered in S.Y. B.Tech. AIDS Department.

SECTION-I

Unit 1: Introduction **(5 Hours)**

Introduction to Computer Graphics- Definition, history and scope of computer graphics, pixel, Pixel dimensions,. Typical graphics pipeline (CPU → GPU), Graphics file formats – raster (BMP, PNG, JPEG, GIF, TIFF), vector (SVG), 3-D (OBJ, FBX, glTF), compression basics, Framebuffer concept, Color Models- RGB, CMYK, HSL,HSV Raster-scan vs. random-scan systems

Unit 2: Output Primitives & Attributes- **(5 Hours)**

Scan-conversion of basic primitives- DDA and Bresenham line algorithms (integer & floating-point forms), Mid-point circle, Polygon rasterisation – scan-line algorithm, inside-outside tests, edge tables, Area-fill algorithms – boundary-fill, flood-fill, seedfill variants, Primitive attributes –intensity mapping, point/line styles, area patterns, Anti-aliasing – super-sampling, area sampling,

Unit 3: 2-D Geometric Transformations & Viewing- **(5 Hours)**

Homogeneous coordinates 3 × 3 transformation matrices, Elementary transforms – translation, rotation, uniform & differential scaling, reflection, shear, Composite transforms; rotation about an arbitrary pivot, Window-to-viewport mapping; aspect ratio preservation, 2-D clipping, Line clipping – Cohen-Sutherland, , Polygon clipping.

SECTION-II

Unit 4: 3D Graphics & Projections- (5 Hours)

3-D Cartesian & homogeneous coordinate systems, 3-D object representations – polygon meshes, quad-meshes, boundary-rep, constructive solid geometry, parametric curves & surfaces (brief), 4 × 4 transformation matrices – translation, rotation about coordinate & arbitrary axes, scaling, reflection, Introduction to Parallel projections – orthographic, axonometric (isometric, dimetric, trimetric), oblique.

Unit 5: Illumination, Shading & Visible-Surface Algorithms- (4 Hours)

Light–material interaction – ambient, diffuse (Lambert), specular (Phong), Shading methods – flat, Gouraud, Phong; normal interpolation, per-fragment shading, texture mapping overview, Shadow generation basics – shadow mapping

Unit 6: Animation & Advanced Applications- (4 Hours)

Principles of animation – key-framing, forward & inverse kinematics Motion capture pipeline, basics of skeletal animation, Morphing techniques – linear blend-shape

List of Practical's:

1. Pixel Plotting & Line Drawing Implement DDA and Bresenham's line drawing algorithms. Compare outputs and discuss accuracy and efficiency.
2. Circle and Ellipse Drawing Implement midpoint algorithms for drawing circles and ellipses using integer arithmetic.
3. Polygon Drawing and Filling Draw arbitrary polygons and implement area-filling algorithms (boundary-fill and flood-fill).
4. 2D Transformations Apply translation, rotation, scaling, and reflection to 2D objects using transformation matrices.
5. Polygon Clipping Implement Sutherland-Hodgman polygon clipping and Cohen-Sutherland line clipping algorithms.
6. Viewport Transformation Demonstrate window-to-viewport mapping with normalized coordinates.
7. 3D Transformations Apply 3D transformations (translation, rotation, scaling) to basic models. Use GenAI (eg. ShapeNet, DreamFusion) to generate the 3D model that can be transformed using learned techniques
8. Projection Techniques Implement orthographic and perspective projections for simple 3D objects.
9. Lighting and Shading Implement flat, Gouraud, and Phong shading models on 3D surfaces.
10. Basic Animation and Object Export Create a key-frame animation sequence and export a 3D object in OBJ format.

List of Tutorials:

Tutorial 1: Understanding Graphics Pipeline and Color Models: Fundamentals of Computer Graphics: Pipeline, Formats, and Color Models (Presentation)

Tutorial 2: Line and Circle Drawing Algorithms

- Draw a line from (2, 2) to (10, 6) using Bresenham's algorithm.
- Use the Midpoint Circle algorithm to draw a circle with radius = 5 centered at (0,0).

Tutorial 3: 2D Geometric Transformations & Viewing

- Translate a triangle with vertices (1,2), (3,5), (5,1) by (2,3).
- Rotate a square (0,0), (0,2), (2,2), (2,0) by 90° about origin.
- Apply Cohen-Sutherland algorithm to clip lines from (-5,5) to (5,7) in the window (- 4,-4) to (4,4).

Tutorial 4: 3D Graphics & Projections

- Rotate a point (1,0,0) about Z-axis by 90°.
- Apply composite transformation (translate → scale → rotate) on a 3D object.

Tutorial 5: Illumination, Shading & Visible Surface Algorithms (Writeup)

- Explain differences using triangular polygon and normal vectors.
- Compare flat, Gouraud, and Phong shading techniques.
- Explain the z-buffer and painter's algorithm.

Tutorial 6: Animation & Advanced Applications

- Animation Techniques and Modern Applications in Computer Graphics (Presentation)

List of Course Projects:

1. Interactive Computer Graphics Learning Toolkit
2. 2D Graphics Editor (Mini Paint Application)
3. Graphics Algorithm Visualizer
4. Raster Graphics Simulator
5. Simple CAD Drawing System
6. Image Format and Compression Explorer
7. 2D Game Graphics Engine
8. Interactive Polygon Editor
9. Digital Map Viewer
10. 3D Projection Visualizer
11. Computer Pipeline simulator
12. 3D Wireframe Modeling System
13. Virtual Architecture Viewer
14. Interactive Lighting and Shading Demonstrator
15. 3D Material and Texture Viewer

16. Texture Mapping Simulator
17. 3D Animated Robot Arm
18. Motion Capture Visualization System
19. 3D Car Driving Simulator
20. Virtual Museum Tour

List of Course Seminar Topics:

1. **Learning tools, visualization of graphics concepts, interactive simulations, educational applications.**
2. **3D Graphics Pipeline and Projection Techniques 3D:** graphics pipeline, homogeneous coordinates, orthographic, axonometric, and oblique projections
3. **Computer-Aided Design (CAD):** Principles and Modern Applications CAD systems, geometric modeling, 3D object creation, engineering applications
4. **3D Object Representation Techniques in Computer Graphics** Polygon meshes, quad meshes, boundary representation (B-Rep), constructive solid geometry (CSG)
5. **Geometric Transformations in 3D Computer Graphics** Translation, rotation, scaling, reflection, arbitrary axis rotation, transformation matrices
6. **Illumination Models in Computer Graphics** Ambient, diffuse (Lambert), specular (Phong), light-material interaction
7. **Shading Techniques for Real-Time Rendering** Flat, Gouraud, and Phong shading, normal interpolation, per-fragment shading
8. **Texture Mapping Techniques and Material Design in 3D Graphics** UV mapping, texture coordinates, procedural textures, material properties
9. **Shadow Mapping Algorithms for Realistic Rendering** Shadow generation, depth maps, shadow mapping pipeline, optimization techniques
10. **Dynamic Lighting in Virtual Environments** Multiple light sources, real-time lighting, global vs. local illumination
11. **Building Virtual Art Galleries Using Modern Graphics Techniques** 3D scene creation, lighting, camera navigation, interactive virtual environments
12. **Animation Principles in Computer Graphics** Key-framing, timing, squash and stretch, anticipation, follow-through
13. **Forward and Inverse Kinematics in Character Animation** Skeletal animation, joint hierarchy, robotic and gaming applications
14. **Motion Capture Technology:** Motion capture pipeline, sensors, data processing, animation retargeting
15. **Skeletal Animation Techniques for Modern Games** Bone hierarchy, skinning, rigging, animation blending
16. **Morphing Techniques in Computer Graphics** Blend shapes, facial animation, image morphing, character expression
17. **Applications of 3D Computer Graphics in Industry** Gaming, films, architecture, healthcare, education, virtual reality
18. **Future Trends in Computer Graphics:** AR, VR, AI, and Digital Twins Emerging technologies,

immersive environments, AI-assisted graphics, Industry 4.0

Text Books: (As per IEEE format)

1. *Donald Hearn & M. Pauline Baker, "Computer Graphics with OpenGL", 4th Edition, Pearson, 2010.*
2. *Edward Angel & Dave Shreiner, "Interactive Computer Graphics: A Top-Down Approach with WebGL", 8th Edition, Pearson, 2020.*
3. *James D. Foley , A. van Dam, John F. Hughes,, et al. "Computer Graphics: Principles and Practice", 3rd Edition, Addison-Wesley, 2019.*

Reference Books: (As per IEEE format)

1. *F. S. Hill Jr. & Stephen Kelley, "Computer Graphics Using OpenGL", 3 rd Edition, Prentice Hall, 2006.*
2. *David F. Rogers & J. Alan Adams, "Mathematical Elements for Computer Graphics", 2nd Edition, McGraw-Hill, 1990.*
3. *Steve Marschner & Peter Shirley (eds.), "Fundamentals of Computer Graphics", 5th Edition, CRC Press, 2021.*
4. *Steven Harrington. Computer Graphics, "A Programming Approach", 2nd Edition, McGraw-Hill,*

Course Outcomes:

The student will be able to –

Upon completion of the course, students will be able to –

1. Explain the architecture, hardware, and applications of computer graphics system
2. Implement raster algorithms for drawing primitives with color and antialiasing.
3. Apply 2D transformations and clipping using matrix operations.
4. Manipulate 3D models and projections for viewing and rendering.
5. Apply shading and visibility techniques to enhance realism.
6. Create simple animations and explore graphics file formats and shaders.

Moocs Links and additional reading material:

https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs90

Future Courses Mapping:

1. Understand Graphics Pipeline & GPU Programming.
2. Implement and simulation of primitives of computer graphics.
3. Geometric Modeling.
4. Projection & Viewing .
5. Illumination & Shading with Real-Time Rendering, Game engines, visualization.
6. Understand Animation Game Development.

Job Mapping:

Graphics Software Engineer, Game Developer, AR/VR Developer, Computer Vision Engineer, UI/UX Designer, CAD Engineer, Animation Programmer, Digital Twin Developer, Metaverse Developer, AI Graphics Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**		**PSO1**	**PSO2**
CO1	2	1	1	-	-	-	-	-	-	-	-		1	1
CO2	1	2	-	-	-	-	-	-	-	-	-		1	1
CO3	2	1	1	-	-	-	-	-	-	-	-		-	1
CO4	-	1	1	-	-	-	-	-	-	-	-		1	1
CO5	2	2	1	-	-	-	-	-	-	-	-		1	1
CO6	2	1	2		3	-	-	-	-	-	-		2	3
Average	1.5	1.3	1	-	0.5	-	-	-	-	-	-		1	1.3

FF No. : 654

HS2002: From Campus To Corporate – 1

Teaching Scheme		Examination Scheme
Credits:	2	CP: NA
Lectures :	2 Hrs/week	CVV:NA
Practical :	0 Hrs/week	MSE(R): 50 Marks
Tutorial :	0 Hrs/week	ESE(R) : 50 Marks

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate,

Employer expectations from fresh graduates, Time management and ownership in corporate settings

Professional Communication Skills: Verbal and non-verbal communication, Email and business writing etiquette, Presentation skills and use of visual aids, Listening skills and telephone etiquette,

Soft Skills and Interpersonal Effectiveness: Body language, grooming, and first impressions, Conflict resolution and negotiation skills, Team dynamics and collaboration, Assertiveness vs. aggressiveness

Resume Building and Job Preparation: Building an effective resume and cover letter, Identifying strengths and achievements, Preparing for technical and HR interviews, Handling rejections and feedback

Group Discussions and Personal Interviews: Group discussion formats and evaluation criteria, Strategies for initiating, contributing, and summarizing, Mock interviews with feedback, STAR technique for answering behavioral questions,

Corporate Etiquette and Workplace Ethics: Meeting and greeting protocol, Dining and social etiquette, Work ethics, punctuality, confidentiality, Respect for diversity and inclusion in the workplace

Adaptability and Emotional Intelligence: Handling pressure, deadlines, and ambiguity, Self- awareness and emotional regulation, Empathy and workplace relationships, Managing feedback and continuous learning,

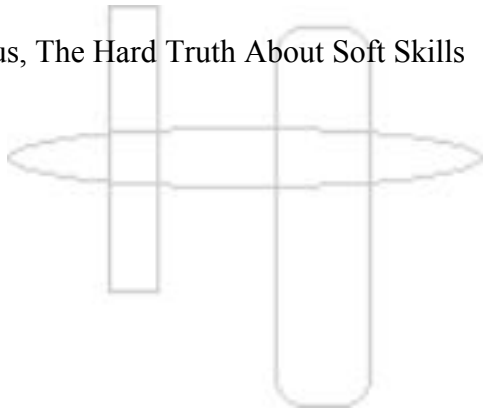
Introduction to Project Management Basics: Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

Faculty are supposed to do conduct following in the class

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions
- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

Books:

1. Dale Carnegie, How to Win Friends and Influence People
2. Stephen R. Covey, 7 Habits of Highly Effective People
3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional
4. Peggy Klaus, The Hard Truth About Soft Skills



FF No.: 654

HS2001: Reasoning & Aptitude Development-3

Teaching Scheme	Examination Scheme
Credits : 1	CP: -- Marks
Lectures : -- Hrs/week	GD/PPT/HA: -- Marks
Practical : -- Hrs/week	MSE (W/O): -- Marks
Tutorial : 1Hrs/week	ESE (W/O): 100 Marks

Prerequisites:

- Basic knowledge of English grammar and vocabulary.
- Fundamental understanding of sentence construction and communication skills.
- Basic arithmetic and mathematical concepts (Class XII level).
- Elementary logical and analytical reasoning skills.
- Familiarity with problem-solving techniques.
- Basic computer literacy and programming concepts.
- Exposure to at least one programming language (C/C++/Java/Python) is desirable.

Course Objectives:

- Develop proficiency in English grammar, vocabulary, reading comprehension, and effective written communication.
- Strengthen logical, analytical, critical, and reasoning abilities to solve diverse aptitude-based problems.
- Apply quantitative and mathematical concepts to solve real-world numerical and analytical problems efficiently.
- Interpret, analyze, and synthesize information from textual, graphical, tabular, and numerical data for informed decision-making.
- Develop computational thinking and programming aptitude through logic building, pseudocode, algorithmic problem solving, and debugging techniques.
- Enhance coding proficiency, optimization skills, and problem-solving abilities to perform effectively in programming assessments, campus recruitment tests, and technical interviews.

Course Outcomes:

After completion of the course, student will be able to

1. **Apply** appropriate English language skills, including grammar, vocabulary, reading comprehension, and effective communication techniques to improve verbal and written proficiency.
2. **Analyze** logical reasoning problems, critical thinking scenarios, and data interpretation tasks to arrive at accurate and evidence-based conclusions.
3. **Solve** quantitative aptitude problems using mathematical concepts, algebraic reasoning, probability, geometry, and analytical techniques in real-life contexts.
4. **Develop** efficient algorithmic solutions by applying computational thinking, programming fundamentals, debugging, optimization techniques, and time complexity analysis to solve coding problems.

SECTION 1
English Language and Communication Skills: English tenses (Present, Past, Future), active and passive voice, sentence correction, parts of speech, syntax, sentence structure, subject–verb agreement, reading comprehension, skimming and scanning, inference, tone, theme, intent identification, contextual vocabulary. **Logical Reasoning and Analytical Thinking:** Deductive reasoning (syllogisms, coding–decoding, statement–conclusion, data sufficiency, directional sense, logical sequencing), inductive reasoning (analogies, classification, number series, sequence completion, trend identification), abductive and critical reasoning (assumptions, arguments, cause–effect, logical evaluation, decision making), information gathering, interpretation and synthesis using graphs, charts, tables and multi-source data.
SECTION 2
Quantitative Aptitude and Mathematical Reasoning: Number systems, integers, fractions, decimals, divisibility, HCF and LCM, prime and rational numbers, algebraic expressions, linear equations and inequalities, proportional reasoning, speed–time–distance, profit and loss, percentages, averages, ages, mixtures, permutations and combinations, probability, geometry, spatial reasoning, pattern recognition and quantitative problem solving. **Programming Aptitude and Automata Skills:** Logic building, pseudocode, programming fundamentals, basic coding problems, program correctness, execution efficiency, intermediate coding challenges, data

structures, algorithms, debugging, optimization techniques, time complexity analysis, industry best coding practices, Automata Pro IT, Automata Pro Max and comprehensive mock assessments.	
Text Books: ***(As per IEEE format)***	
1	R. S. Aggarwal, *Quantitative Aptitude for Competitive Examinations*, Revised and Enlarged Edition. New Delhi, India: S. Chand and Company Ltd., 2017.
2	R. S. Aggarwal, *A Modern Approach to Verbal & Non-Verbal Reasoning*. New Delhi, India: S. Chand Publishing.
3	Dr. M. B. Lal and A. Gupta, *CSAT Logical Reasoning & Analytical Ability*. Agra, India: Upkar Prakashan.
4	G. Laakmann McDowell, *Cracking the Coding Interview: 189 Programming Questions and Solutions*, 6th ed. Palo Alto, CA, USA: CareerCup, 2015.
Reference Books: ***(As per IEEE format)***	
1	B. H. Dowden, *Logical Reasoning*. California State University, Sacramento, CA, USA, 2011.
2	*501 Challenging Logic and Reasoning Problems*, 2nd ed. New York, NY, USA: LearningExpress, LLC, 2005.
3	*Logic and Reasoning Problems*. LearningExpress Practice Series.

CO – PO / PSO Articulation Matrix

CO	Program Outcome (PO)												
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO 1**	**PSO 2**
CO1	–	–	–	–	–	–	–	–	3	–	2	-	-
CO2	2	2	–	–	–	–	–	–	1	–	2	-	-
CO3	2	2	–	–	–	–	–	–	1	1	2	2	-
CO4	2	2	3	3	3	–	1	–	1	1	3	-	3
Avg	**2**	**2**	**3**	**3**	**3**	**–**	**1**	**–**	**1.5**	**1**	**2.25**	2	3

FF No.: 654

AI2305: Design Thinking III

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	0 Hrs/week	ESE: 100 Marks
Tutorial :	1 Hrs/week	

Course Prerequisites:

Basic knowledge of research work, research paper and patent.

Course Objectives:

1. Understand the concepts of design thinking approaches
2. Apply both critical thinking and design thinking in parallel to solve problems
3. Apply some design thinking concepts to their daily work
4. To provide ecosystem for students and faculty for paper publication and patent filing

Course Relevance:

The course is offered in S.Y. B.Tech. to all branches of Engineering

SECTION-I

Contents for Design Thinking:

Structure of the paper Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools

Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search Filing of patent Writing answers to reviewer questions

Modification in manuscript Checking of publication draft

Assessment Scheme:

Publication of paper or patent

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Understand the importance of doing Research

CO2: Interpret and distinguish different fundamental terms related to Research

CO3: Apply the methodology of doing research and mode of its publication

CO4: Write a Research Paper based on project work

CO5: Understand Intellectual property rights

CO6: Use the concepts of Ethics in Research

CO-PO Mapping:

	Program Outcomes (PO)											PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	1	2	2	1	3	1	3	2	2	2	2	2
CO2	2	2	3	3	2	2	2	3	2	2	1	2	3
CO3	2	2	2	2	2	2	2	3	2	2	3	2	3
CO4	2	2	2	2	2	2	1	3	2	2	3	2	3
CO5	2	2	2	2	2	2	2	3	2	2	3	3	2
CO6	2	2	2	2	2	2	2	3	2	2	3	3	2
Average	2	1.83	2.16	2.16	1.83	2.16	1.66	3	2	2	2.5	2.33	2.50

FF No.: 654

AI 2306: Engineering Design & Innovation III

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	4 Hrs/week	MSE(R) : 30 Marks
Tutorial :	0 Hrs/week	ESE(R) : 70 Marks

Course Prerequisites:

Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, the curriculum can be enriched with semester-long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy, and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question,

challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-I

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Moocs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

Upon completion of the course, student will be able to –

CO1: Identify the real life problem from a societal need point of view
CO2: Choose and compare alternative approaches to select the most feasible one
CO3: Analyse and synthesize the identified problem from a technological perspective

CO4: Select the best possible solution to solve the problem.
CO5: Design & Develop a working model of the proposed solution.
CO6: Testing and validating product performance

Future Courses Mapping:

Major Project

Job Mapping:

Job opportunities that one can get after learning this course

Software Engineer. Software Developer, IT Engineer, Research Associate.

CO - PO Mapping:

	Program Outcomes (PO)												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	3	2			3				2	2		2
CO2				3	2	2		2				2	2
CO3	2		3		3			2	2			2	2
CO4		2				3						2	2
CO5			3				2		2			3	
CO6		2			3				2	3		2	
Average	0.83	1.16	1.33	0.5	1.33	1.33	0.33	0.66	0.5	0.83	0.33	1.83	1.33

FF No: 654

AI2307: Object Oriented Programming

Teaching Scheme		Examination Scheme
Credits:	4	CP: 20 Marks
Lectures :	3 Hrs/week	Pract+CVV:40 Marks
Practical :	2 Hrs/week	ESE : 40 Marks
Tutorial :	0 Hrs/week	

Course Prerequisites:

1. Fundamentals of Computer programming and problem solving.

Course Objectives:

1. Understand the principles and concepts of Object-Oriented Programming.
2. Develop software solutions using C++ and Java.
3. Apply object-oriented principles to design reusable and maintainable applications.
4. Implement exception handling, file handling, templates, collections, multithreading, and GUI programming.
5. Develop problem-solving and programming skills required for AI, data science, and software development applications using using object-oriented design.

Course Relevance:

The course Object-Oriented Programming provides AI&DS students with a strong programming foundation. It teaches modular, reusable, and efficient coding through core OOP concepts. These skills are essential for developing AI models, data structures, and performance-critical systems.C++ is used in many AI libraries and frameworks like TensorFlow and OpenCV. The course also enhances algorithmic thinking and prepares students for research and industry roles.

SECTION-I

Introduction to C++ and OOP Concepts: Evolution of Programming Paradigms, Procedure-Oriented Programming vs Object-Oriented programming, Principles of OOP, Applications of OOP, Structure of a C++ Program, Standard Namespace, User-defined Namespace, Tokens, Keywords and Identifiers, Variables, Constants and Data Types, Operators and Expressions, Input/Output Streams, Control Statements, Functions, Function Overloading, Inline Functions, Default Arguments**. (03 Hrs)**

Constructors and Destructors: Default Constructor, Parameterized Constructor, Copy Constructor, Dynamic Constructor, Constructor Overloading, Destructor. **(02 Hrs)**

Inheritance: Need for Inheritance, Types of Inheritance, Visibility Modes/ Access specifiers, Constructors

in Derived Classes **(02 Hrs)**

Operator Overloading: Need for Operator Overloading, Unary Operator Overloading, Binary Operator Overloading, Rules for Operator Overloading, Overloading using Member and Friend Functions **(02 Hrs)**

Runtime Polymorphism: Function Overriding, Virtual Functions, Pure Virtual Functions, Abstract Classes, Early Binding and Late Binding.

Pointers and Dynamic Memory: Pointers: Basics, Pointer to Object, this Pointer. Memory Allocation new and delete Operators, Dynamic Objects,Array of Objects Using Pointers, this Pointer. **(03 Hrs)**

Templates: Function Templates, Class Templates, Template Specialization

Exception Handling: try, throw and catch, Multiple Catch Blocks, Nested try Blocks, User-Defined Exceptions. **(02 Hrs)**

SECTION-II

File Handling: Text Files, Binary Files, Random Access Files, File Modes

Standard Template Library: Introduction to STL, Containers, Vector, List, Set, Map, Iterators, Function Objects (Functors) **(03 Hrs)**

Introduction to Java: Features of Java, JVM, JRE and JDK, Java Program Structure, Data Types, Operators, Control Statements, Arrays, Command Line Arguments **(02 Hrs)**

Object-Oriented Programming in Java: Classes and Objects, Constructors, Methods, Method Overloading, this and super Keywords, Packages, Access Modifiers **(02 Hrs)**

Inheritance and Polymorphism: Inheritance, Method Overriding, final Keyword, Abstract Classes, Interfaces. **(02 Hrs)**

Exception Handling: Exception Hierarchy, try-catch-finally, throw and throws, User-defined Exceptions

Multithreading: Thread Class, Runnable Interface, Thread Life Cycle, Thread Synchronization

Generic Programming: Generic Classes, Generic Methods **(02 Hrs)**

GUI Programming: Swing Components, Event Handling, Layout Managers **(03 Hrs)**

List of Practical's: (Any 12)

1.Write a C++ program to implement a class named Book with the following data members:

Private:bookId, bookName, author, price

Public:getDetails(), printDetails()

The program should accept details of multiple books from the user, display the book information in tabular form, and calculate and display the total price of all the books.

2.Student Records System

Write a C++ program to implement a class named Student with the following data members:

Private:

studentId, studentName, department, marks

Public:

getDetails(), displayDetails()

The program should accept details of multiple students from the user, display the student information in a tabular format, and calculate and display the average marks of all students.

3.Employee Payroll System

Write a C++ program to implement a class named Employee with the following data members:

Private:

empId, empName, designation, salary

Public:

acceptData(), showData()

The program should accept details of multiple employees from the user, display the employee information in a table format, and calculate and display the total salary expenditure of the organization.

4.Write a C++/JAVA program to implement a class Book with suitable data members. Use constructors to initialize book details and a destructor to indicate object destruction, thereby demonstrating the concepts of objects, classes, constructors, and destructors.

5.Write a C++ program to implement a class named Student with suitable data members such as roll number, name, and marks.

Use:

• A parameterized constructor to initialize student details

• A destructor to display a message when a Student object is destroyed

The program should create multiple Student objects to demonstrate the working of constructors and destructors and the concept of object creation and destruction.

6.Write a C++/JAVA program to implement a class named Employee with suitable data members such as employee ID, name, and salary.

Use:

• A constructor to initialize employee information

• A destructor to indicate when an Employee object is removed from memory

The program should create and display details of multiple Employee objects, thereby illustrating object-oriented principles, constructors, and destructors.

7. Design and implement a C++ program to model a banking system where the class BankAccount keeps track of the total number of accounts created and the total balance across all accounts. Use static data members to maintain these values and static member functions to retrieve and display them. Additionally, implement a member function in the BankAccount class that takes another BankAccount object as an argument and displays the two account balances in ascending order.

8. Write a C++ program to create a Student class with the following:

Private data members: Roll Number, Name, Marks of five subjects, Percentage, Result Division

Public members: Parameterized constructor, destructor, functions to calculate percentage, determine grade/division, and display student details.

Implement a friend function comparePercentage() that takes two Student objects and displays the one with the higher percentage.

9. Design and implement a C++ program to implement a ParkingTicket class with:

Private data members:

ticketNumber, vehicleNumber, parkingFee

Static data members:

totalTickets, totalCollection

Use:

• Constructors to initialize ticket details

• Static member functions to display total tickets issued and total collection

Dynamically allocate ParkingTicket objects using new.

Also, define a friend functioncompareFees() that accepts two ParkingTicket objects and displays their parking fees in increasing order.Release allocated memory using delete.

10. Design a C++ class Complex with data members to store the real and imaginary parts. Provide default and parameterized constructors.Implement operator overloading for +, -, and * to perform arithmetic operations on complexnumbers.Write a program that creates two complex objects and displays the results of these operations.

11.Create a class Number that stores an integer value.

- Provide member functions to input and display the value.
- Overload the unary minus operator (-) so that it returns a new object whose value is the negative of

the original number.

- Demonstrate the operator by showing both the original and the negated value.

12. Design a class Integer that stores an integer value.

- Provide suitable constructors to initialize the value.
- Overload both increment operators:
- Pre-increment (++obj) to increase the value and return the updated object.
- Post-increment (obj++) to return the original value before incrementing.
- Write a program that creates an object of class Integer and demonstrates both operations.
- Display the value of the object:
- before increment
- after pre-increment
- after post-increment

13. Design and implement a C++ program to demonstrate inheritance using the following employee hierarchy:

Base Class: Employee with data members: Emp_name, Emp_id, Address, Mail_id, and Mobile_no.

Derived Classes: Programmer, TeamLead, AssistantProjectManager, and ProjectManager. Each derived class should have an additional data member Basic Pay (BP).

Salary components for all employees are calculated as:

Dearness Allowance (DA): 52% of BP

House Rent Allowance (HRA): 27% of BP

Provident Fund (PF): 12% of BP

Staff Club Fund: 0.1% of BP

Implement member functions to calculate gross and net salary and generate pay slips for all employees.

14. Problem Statement: Employee Salary System Using Single Inheritance

Design a C++ program to manage employee information in an organization.

1. Create a base class Employee that contains:

- Employee ID
- Employee name
- A function to input and display employee details

2. Create a derived class Manager that extends the Employee class and contains:

- Monthly bonus
- A function to calculate and display total salary (basic salary + bonus)

3. In the main() function:

- Create an object of the Manager class
- Accept employee details and bonus
- Display complete employee information along with total salary

15. Design a base class Shape with two double-type data members to represent the dimensions of the shape. Include member functions to input the dimensions and a pure virtual function compute_area() for calculating the area.

Derive two classes: Triangle and Rectangle from Shape and override the compute_area() function to calculate the area according to the specific shape.

Write a program that:

a) Accepts the dimensions of a triangle or rectangle from the user.

b) Uses dynamic binding to call the appropriate compute_area() function.

c) Displays the calculated area of the shape.

16. Electricity Bill System

Create an abstract base class ElectricityBill with:

- Two double data members: unitsConsumed and ratePerUnit.
- A function input() to accept values.
- A pure virtual function calculate_bill().

Derive two classes:

- Domestic → Bill = units × rate
- Commercial → Bill = units × rate + 10% surcharge

Write a C++ program to:

a) Accept input from the user.

b) Use dynamic binding to calculate the correct bill.

c) Display the final bill amount.

17. Design and implement an interface for Vehicles to represent common functionalities. Consider vehicles such as Bicycle, Bike, and Car, which share the following functionalities:

- gearChange()
- speedUp()
- applyBrakes()

Create an interface Vehicle that declares these methods. Implement the interface in the classes Bicycle, Bike, and Car, providing custom implementations for each method according to the specific vehicle type.

18. Student Result Processing System

Create a C++ program to calculate the percentage and grade of students.

The program should:

Accept marks for multiple subjects

Calculate total, percentage, and grade

Use exception handling for:

Marks entered below 0 or above 100

Division by zero while calculating percentage

19. Create a class using JAVA AreaCalculator with overloaded methods area() to:

Calculate area of a square

Calculate area of a rectangle

Calculate area of a circle

20.Design and implement a Java application to calculate the perimeter of different geometric shapes—Circle, Rectangle, and Triangle—using hierarchical inheritance.

Create a base class Shape that defines a general method perimeter(). Then derive three classes Circle, Rectangle, and Triangle from Shape. Each derived class must override the perimeter() method to compute the perimeter based on its specific formula.

The program should accept input values for each shape, create corresponding objects, and display the calculated perimeters using runtime polymorphism (base class reference).

21.Write a Java program to create two threads using the Runnable interface & Thread class. One thread should display even numbers from 1 to 20, and the other should display odd numbers from 1 to 20.

22.Write a Java program to create two threads. One thread should display even numbers between 1 and 50, and the other thread should display numbers divisible by 5 between 1 and 50. Use multithreading concepts in Java.

List of Tutorials:

1. Write a Program C++ Program to Make a Simple Calculator to Add, Subtract, Multiply or Divide Using switch...case.
2. Write a Program C++ Program to Display Prime Numbers Between Two Intervals Using Functions.
3. Write a Program C++ program to Reverse a Sentence Using Recursion.
4. Write a Program C++ Program to Add Two Matrices Using Multi-dimensional Arrays.

5. Write a Program C++ Program to Multiply Two Matrix Using Multi-dimensional Arrays
6. Write a Program C++ Program to Store and Display Information Using Structure.
7. Write a Program that computes the simple interest and compound interest payable on principal amount (in Rs.) of loan borrowed by the customer from a bank for a given period of time (in years) at specific rate of interest. Further determine whether the bank will benefit by charging simple interest or compound interest.
8. Write a program to illustrate function overloading. Write 2 overloading functions for adding two numbers.
9. Write a java program to print odd and even numbers from an array Create a random sentence generator by stitching together subject, verb, and object arrays—like a primitive text generator.

List of Course Projects:

1. Student Management System Add, delete, modify student records Store data using file handling
2. Bank Management System Simulate account creation, withdrawal, deposit Basic file I/O for saving account info
3. Library Management System Add/search/delete books Manage user borrowing history
4. Quiz Game Simple command-line quiz Score calculation and question randomization
5. Tic-Tac-Toe Game Two-player mode Command-line interface
6. Basic Calculator Perform +, −, ×, ÷ Extend to scientific mode
7. Hospital Management System Manage patient records, doctor schedules Use structs or classes for data organization
8. Inventory Management System Track products, quantity, price File/database storage (basic)
9. Employee Management System Add/remove employees Track salary, department, attendance
10. Simple Chat Application (Console-based) Simulate chat between users (offline using files or in-memory Use object-oriented design
11. Snake Game (Console Version) Use characters to simulate game environment Control snake movement with keyboard

List of Course Seminar Topics: (*if applicable to your course*)

1. Object-Oriented Programming Paradigms and Their Role in AI Development
2. Evolution of C++ in AI and Machine Learning Frameworks
3. Designing Intelligent Systems Using Classes and Objects in C++
4. Constructor and Destructor Management in AI System Design
5. Operator Overloading in C++: Applications in AI Vector and Matrix Computations
6. Friend Functions and Classes in AI: Use Cases and Best Practices
7. Hierarchical and Hybrid Inheritance in AI Model Architecture
8. Polymorphism in AI Evaluation Systems: Compile-time vs Runtime Behavior

9. Dynamic Memory Allocation for Large-Scale AI Data Handling

10. Exception Handling in C++ for AI Application Reliability

11. Generic Programming with Templates in AI Frameworks

12. Standard Template Library (STL) and Its Applications in AI Dataset Management

13. Building Lightweight AI Libraries in C++: Challenges and Opportunities

14. Comparison of C++ and Python for AI Development: A Performance Perspective

15. Memory Optimization Techniques in C++ for AI Algorithms

Text Books: (As per IEEE format)

1. B. Stroustrup, The C++ Programming Language, 4th ed. Boston, MA, USA: Addison-Wesley, 2013.

2. H. Schildt, C++: The Complete Reference, 4th ed. New York, NY, USA: McGraw-Hill Education, 2003.

3. E. Balagurusamy, Object-Oriented Programming with C++, 8th ed. New Delhi, India: McGraw-Hill Education, 2020.

Reference Books: (As per IEEE format)

1.S. B. Lippman, J. Lajoie, and B. E. Moo, C++ Primer, 5th ed. Boston, MA, USA: Addison-Wesley, 2012.

2.R. Lafore, Object-Oriented Programming in C++, 4th ed. Indianapolis, IN, USA: SAMS Publishing, 2001.

3.M. A. Weiss, Data Structures and Algorithm Analysis in C++, 4th ed. Boston, MA, USA: Pearson, 2014.

4. J. R. Hubbard, Programming with C++, 2nd ed. New York, NY, USA: McGraw-Hill Education, 2000.

Course Outcomes:

The student will be able to –

CO1: Explain the fundamental concepts of Object-Oriented Programming and develop C++ programs using classes, objects, functions, and basic programming constructs.

CO2: Implement object-oriented features including constructors, inheritance, operator overloading, and runtime polymorphism to develop reusable C++ applications.

CO3: Develop C++ programs using dynamic memory allocation, templates, exception handling, namespaces, and file handling techniques.

CO4: Apply the Standard Template Library (STL) and C++ programming concepts to design efficient and modular software solutions.

CO5: Develop Java applications using object-oriented principles including classes, inheritance, interfaces,

packages, exception handling, and dynamic method dispatch.

CO6: Design and implement Java applications incorporating multithreading, collections, generics, and GUI programming, to solve real-world problems.

Moocs Links and additional reading material: (If any)

1. Programming using Java| Java Tutorial | By Infosys Technology
https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01304972186110361645_shared/overview
2. An Introduction to Programming through C++ – Prof A.G. Ranade- NPTEL- computer science and engineering – NOC https://nptel.ac.in/courses/106/101/106101208/#

Future Courses Mapping:

1. Artificial Intelligence / Machine Learning
2. Advanced Data Structures, Advanced Java, Spring Framework, Grails Framework

Job Mapping: (Optional)

1. Technical Support Engineer (C++ based software)
2. Software Developer / Programmer (C++ based software)
3. Java Programmer, Application Developer, Design Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**		**PSO1**	**PSO2**
CO1	3	1			2									
CO2	3	2	2	1	2								2	
CO3	3	3	3	2	2				1				2	1
CO4	3	2	2	2	2				1					1
CO5	3	3	3	2	3				2				3	2
CO6	3	2	2	3	3				2	1	1		3	3
Average	3	2.16	2.4	2	2.33				1.5	1	1		2.5	2.33

FF No.: 654

AI 2308: Computer Network

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	Lab Assessment: 10 Marks
Tutorial :	0 Hrs/week	ESE (W) : 40 Marks

Course Prerequisites: Fundamentals of Computer, C/C++ programming

Course Objectives:

1. Understand the importance of Computer Network and their usage.
2. Study error control and flow control techniques.
3. Solve real-world problems in the context of today's internet (TCP/IP and UDP/IP).
4. Distinguish and relate various physical Media, interfacing standards, and adapters.
5. Implement mathematically and logically the working of computer protocols in the abstract.

Course Relevance:

This course will show you the applications, communications protocols, and network services that make a computer network work. We will closely follow the top-down approach to computer networking, which will enable you to first understand the most visible part i.e. the applications, and then seeing, progressively, how each layer is supported by the next layer down. Most of the time, our example network will be the Internet.

SECTION-I

Unit 1: Introduction to Computer Network: (4 Hours)

Introduction to computer network.
Type of Computer Network : LAN, MAN, WAN, PAN, Ad-hoc Networks.
Network Topologies : Bus, Ring, Tree, Star, Mesh, Hybrid
Transmission media- Guided media, unguided media.
Transmission Modes- Simplex, Half-Duplex and Full-Duplex.
Network Devices- Hub, Repeater, Bridge, Switch, Router, Gateways and Router
Network Models: OSI Model, TCP/IP Model.

Unit 2: Data Link Layer: (5 Hours)

Data Link Layer: Data Link Layer Services,
Error Detection and Correction: Linear Block Codes: hamming code, Hamming Distance, parity

check code. Cyclic Codes: CRC (Polynomials), Internet Checksum.
Framing: fixed-size framing, variable size framing.
Flow control Protocols: Stop-and-Wait Automatic Repeat Request (ARQ), go-back-n ARQ, Selective repeat ARQ.
Random Access Techniques: ALOHA,CSMA, CSMA/CD, CSMA/CA.

Unit 3: Network Layer: (5 Hours)

Network Layer: Network Layer Services.
IPv4 Addresses: Static and Dynamic Configuration Classful and Classless Addressing, Special Addresses, Subnetting, Super-netting, Delivery and Forwarding of IP Packet, NAT (Network Address Translation).
IPv4: Datagram's, Fragmentation, Options, Checksum.
IPv6: Addressing: Notations, Address Space, Packet Format

SECTION-II

Unit 4: Network Layer Routing Protocols: (5 Hours)

Routing: Optimality Principle, Static vs Dynamic Routing Tables, Routing Protocol: Intra and Inter Domain Routing.

Unicast Routing Algorithms: Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Path State Routing.

Interior Routing algorithms: RIP, EIGRP, OSPF

Exterior Routing Algorithms: BGP

Unit 5: Transport Layer: (5 Hours)

Transport layer: Transport layer services & Duties,

Transport Control Protocol: TCP header, Connection Establishment, Flow control,

Congestion Control: Leaky Bucket, Token Bucket, Load Shedding and TCP Timers.

User Datagram Protocol: UDP header, Datagram, Services, Applications,

Socket: Primitives, TCP & UDP Sockets.

Unit 6: Application Layer: (4 Hours)

Client Server Paradigm: Communication using TCP and UDP, Peer to Peer Paradigm, Application Layer Protocols: DNS, FTP, HTTP, SMTP, POP, IMAP, MIME.

List of Practical's: (Any Six)

1. Study of the following connectivity test tools with their options –
   - ifconfig, arp, route, traceroute
   - nmap, netstat, finger
2. Network Traffic Analysis using AI.

- PyShark/Wireshark (for traffic capture)

3. Write a program using a TCP socket
   - Send Hello Message
   - File transfer
4. Write a program using UDP Sockets to enable file transfer (Script, Text, Audio, and Video on file each) between two machines.
5. Write a Program to implement Framing methods.
6. Write a Program to implement Error detection and Correction
   - CRC code
   - Hamming code
7. Write a program to find the class and type of a given IP address.
8. Write a program to demonstrate subnetting and find the subnet masks.
9. Write a Program to implement the shortest path routing algorithm.
10. Write a Program to implement the sliding window protocol.
    - Go Back N
    - Selective Repeat

List of Tutorials:

1. Identification of various network components
2. Establishing LAN
3. Installation of network device drivers
4. Use/installation of proxy server
5. Configuration of network devices in CISCO packet tracer (Windows/Linux)
6. Implement communication between various network devices using CISCO packet tracer (Windows/Linux)

Network traffic monitoring using Wireshark/Ethereal (Windows/Linux)

List of Course Projects:

1. Write a program using TCP sockets for wired networks to implement
   a. Peer-to-Peer Chat b. Multi User Chat Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer-to-peer mode.
2. Implementation of shortest path protocol
3. Implementation of string encryption and decryption
4. Implementation of character stuffing and de-stuffing
5. Execution and analysis of Network commands
6. To find out details of the network from the IP addressing scheme using the ‘C’ code
7. Implement real-time Internet route optimization.
8. Implement Broadcast Server System.
9. Implement a real-time voting System.
10. Real-time packet capture and analysis for malware in wireless networks.

Text Books: (As per IEEE format)

1. *James F. Kurose, and Keith W. Ross," Computer Networking: A Top-Down Approach," 4th edition, Publisher: Addison-Wesley ISBN: 0-321-49770-8*

2. *Behrouz A. Forouzan ,"Data Communication and Networking", 4th edition, Tata McGraw Hill Andrew S. Tanenbaum, "Computer Networks" ,5th Edition, Pearson Education*

Reference Books: (As per IEEE format)

1. Kurose, Ross, "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204
2. Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Network", Wiley, ISBN: 0-470-09510-5
3. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols", Prentice Hall, 2004

Course Outcomes:

Upon completion of the course, student will be able to –

1. Select network architecture, topology and essential components to design computer networks.
2. Design mechanisms to demonstrate channel allocation in wired and wireless computer networks.
3. Estimate reliability issues based on error control, flow control by using bandwidth, latency, throughput and efficiency.
4. Implement the client server application using socket.
5. Develop Client-Server architectures and prototypes by the means of correct standards, protocols and technologies.
6. Analyse data flow between peer-to-peer in an IP network using Application, Transport and Network Layer Protocols.

Moocs Links and additional reading material:

1. https://nptel.ac.in/courses/106105183
2. https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/

https://www.my-mooc.com/en/categorie/computer-networking

Future Courses Mapping:

1. **Network Security**
2. **Cybersecurity**
3. **Software-Defined Network**

Job Mapping:

Job opportunities that one can get after learning this course

1. **Network Administrator**
2. **System Engineer**
3. **Network Architect**

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**
CO1	1	3	2	3	2	-	-	-	-	-	-		2	1
CO2	2	3	3	3	2	2	-	-	-	-	-		2	1
CO3	3	3	3	2	2	2	-	-	-	-	-		3	-
CO4	2	2	3	-	3	-	-	-	2	1	-		3	1
CO5	3	2	3	-	3	-	-	-	2	1	-		2	2
CO6	3	3	2	2	2	-	-	-	-	-	-		2	1
Average	2.3	2.3	2.6	1.6	2.3	0.6	-	-	0.6	0.3	-		2.3	1

FF No.: 654

AI 2309: Mathematical Foundation of Artificial Intelligence

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	Lab Assessment: 10 Marks
Tutorial :	0 Hrs/week	MSE(W): 40 Marks

Course Prerequisites:

Discrete Mathematics, Basic calculus and algebra, Fundamentals of probability and statistics, Programming basics

Course Objectives:

1. Understand the core mathematical concepts essential for Artificial Intelligence.
2. Learn and apply vector, matrix, and linear equation solving techniques.
3. Understand how to represent and manipulate knowledge using propositional and first-order logic.
4. Learn to apply inference methods for logical reasoning and problem-solving.
5. Understand and apply probabilistic reasoning techniques to handle uncertainty in AI.

Course Relevance:

This course builds core math and logic skills essential for AI, preparing students for advanced studies and real-world intelligent systems.

SECTION-I

Unit 1: Vectors and Matrix Operations **(5 Hours)**

Introduction to Mathematical Foundations for AI - Overview of AI applications, review of key mathematical areas: linear algebra, probability, optimization.

Vectors and Vector Spaces - Definition of vectors, vector operations, dot product, linear combinations, basis, dimension, and orthogonality.

Matrices and Matrix Operations - Matrix addition, multiplication, transpose, inverse, and special matrices (identity, diagonal, orthogonal), applications in transformations.

Unit 2: System of Linear Equations and Neural Network introduction **(5 Hours)**

Systems of Linear Equations - Solving linear systems using Gaussian elimination and matrix inversion, rank and consistency of solutions, introduction to LU decomposition. Introduction and role of Neural Network(NN), Biological Neural Network vs Artificial Neural Network, Mc Culloch Pitts NN, Perceptron, Activation functions

Unit 3: Introduction to Calculus and Derivatives (4 Hours)

Essence of calculus, integrals and derivatives, Derivation rules: Power rule, product rule and chain rule, Euler's number and continuous growth, integral intuition, definite and indefinite integrals, calculus in machine learning

SECTION-II

Unit 4: First Order Logic (5 Hours)

Syntax & semantics of First order logic- Models for first-order logic, Symbols and interpretations, Atomic & complex sentences, Quantifiers, Equality, Assertions and queries in first-order logic, The wumpus world
Knowledge-engineering in First order logic- knowledge-engineering process, electronic circuits domain

Unit 5: Inference in First-Order Logic (5 Hours)

Inference rules for quantifiers, Reduction to propositional inference, Unification and Lifting- first order inference rule, Unification, Storage and retrieval ,Forward Chaining - First-order definite clauses, forward-chaining algorithm, Efficient forward chaining, Backward Chaining- backward-chaining algorithm, Logic programming, Database semantics of Prolog, Resolution - Conjunctive normal form for first-order logic, The resolution inference rule, Completeness of resolution , Resolution strategies

Unit 6 : Propositional Logic (4 Hours)

Introduction to propositions and truth values, use of logical connectives and truth tables, concepts of equivalence and laws of logic, identification of tautology, contradiction, and contingency, conversion to CNF and DNF, application of inference rules and proof techniques, and use of propositional logic in AI-based reasoning and decision-making.

List of Practical's:

1. Implement basic vector and matrix operations in C.
2. Implement solution of linear equations in C.
3. Compute eigenvalues and eigenvectors of a matrix in C using the characteristic polynomial method.
4. Create a knowledge base with facts, rules, and quantifiers; perform queries using assertions and equality.(Prolog)
5. Represent and query using first order logic in Prolog.
6. Implement forward chaining in Prolog for a given set of first-order definite clauses.
7. Implement backward chaining in Prolog to answer queries from a knowledge base.
8. Represent and evaluate logical statements using Prolog facts and rules; verify truth values for given propositions.
9. Implement Prolog rules to convert logical expressions into CNF and DNF forms.

10. Represent uncertain knowledge in Prolog using probabilistic facts and rules; perform queries to calculate probabilities.
11. Implement a simple Bayesian network in Prolog for a given domain and perform inference on it.
12. Implement an Expert System in Prolog (any chosen domain) using rules and inference to suggest possible outcomes based on user input. (AI Base assignment)

List of Course Projects:

1. Expert System for Medical Diagnosis in Prolog – Knowledge base with rules and inference to suggest diagnoses based on symptoms.
2. Wumpus World AI Agent – Implement an intelligent agent in Prolog to navigate the Wumpus World using first-order logic inference.
3. Bayesian Network for Weather Prediction – Build and query a probabilistic model in Python to predict weather conditions.
4. AI-based Student Performance Predictor – Use Python with NumPy/Pandas to analyze student data and predict performance using probabilistic reasoning.
5. Matrix-based Image Transformation Tool – Implement rotation, scaling, and translation operations on images using matrix operations in C/Python.
6. Automated Reasoning System – Implement forward and backward chaining inference in Prolog for a custom domain (e.g., troubleshooting electronics).
7. PCA-based Dimensionality Reduction – Apply Principal Component Analysis on a dataset to reduce dimensions and visualize results.
8. Traffic Violation Detection System – Knowledge-based AI model in Prolog to detect rule violations from event data.
9. AI-powered Quiz Game – Prolog or Python-based reasoning engine that generates and answers logical queries dynamically.
10. Disease Outbreak Prediction Model – Bayesian inference model to estimate the probability of outbreak spread using given datasets

Text Books: (As per IEEE format)

1. M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
2. G. Strang, *Linear Algebra and Its Applications*, 5th ed. Boston, MA, USA: Cengage Learning, 2016.
3. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
4. Townsend Carl , “Introduction to Turbo Prolog”

Reference Books: (As per IEEE format)

1. S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 5th ed., Amsterdam, Netherlands: Academic Press, 2014.
2. D. P. Bertsekas and J. N. Tsitsiklis, Introduction to Probability, 2nd ed., Belmont, MA, USA: Athena Scientific, 2008.

3. S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge, U.K.: Cambridge Univ. Press, 2004.

Moocs Links and additional reading material:
https://onlinecourses.nptel.ac.in/noc25_cs136/preview

Course Outcomes:
Upon completion of the course, the student will be able to –

1. Apply vector, matrix, and transformation concepts to solve AI-related mathematical problems.
2. Solve linear equations and compute eigenvalues/eigenvectors for AI and stability analysis.
3. Use first-order logic syntax, semantics, and knowledge-engineering for problem representation and reasoning.
4. Apply inference methods—unification, chaining, and resolution—for logic-based problem solving.
5. Use propositional logic and inference techniques for AI reasoning and decision-making.
6. Apply probabilistic reasoning and Bayesian networks to handle uncertainty in AI systems.

CO - PO Mapping:

	Program Outcomes (Pos)												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	1											
CO2	3	1	1										
CO3		2			2							1	
CO4		2											
CO5		2											
CO6	1		2	1									2
Average	1.00	1.33	0.5	0.17	0.33							0.17	0.33

FF No.: 654

AI2310: Probability And Applied Statistics

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	LA: 10 Marks
Tutorial :	0 Hrs/week	ESE: 40 Marks

Course Prerequisites:

1. Basic Mathematics

Course Objectives:

1. Explain the basic concepts of probability and statistics.
2. Apply probability concepts to solve engineering problems.
3. Calculate statistical measures for data analysis.
4. Use probability distributions to model real-world situations.
5. Apply statistical methods for hypothesis testing and decision-making.
6. Interpret data using statistical techniques and software tools.

Course Relevance:

This course is offered to Second Year B.Tech (Artificial Intelligence & Data Science) students. It provides a foundation in probability and applied statistics for analyzing uncertainty, interpreting data, and making informed decisions. The course equips students with statistical techniques essential for Artificial Intelligence, Machine Learning, Data Science, Business Analytics, and other engineering applications.

SECTION-I

UNIT I: Probability Theory (04 Hours)

Introduction to Probability, Sample Space and Events, Classical, Empirical, and Axiomatic Approaches to Probability, Addition and Multiplication Theorems of Probability, Conditional Probability, Independent Events, Bayes' Theorem, and Engineering Applications of Bayes' Theorem.

UNIT II: Random Variables and Probability Distributions (06 Hours)

Discrete and Continuous Random Variables, Probability Mass Function (PMF), Probability Density Function (PDF), Cumulative Distribution Function (CDF), Mathematical Expectation, Linearity of Expectation, Expectation of Independent Random Variables, Joint Probability Distribution, Discrete Probability Distributions (Binomial and Poisson Distributions), and Continuous Probability Distribution (Normal Distribution).

UNIT III: Continuous Probability Distributions (04 Hours)

Uniform Distribution, Log-Normal Distribution, Beta Distribution, Gamma Distribution, Properties of Continuous Probability Distributions, and Engineering Applications.

SECTION-II

UNIT IV: Descriptive Statistics and Data Presentation (06 Hours)

Introduction to Statistics and its Engineering Applications, Types of Data (Qualitative and Quantitative), Data Collection Methods, Frequency Distribution and Data Tabulation, Graphical Representation of Data (Bar Charts, Histograms, Pie Charts, and Box Plots), Measures of Central Tendency (Mean, Median, and Mode), Measures of Dispersion (Range, Variance, Standard Deviation, and Coefficient of Variation), and Measures of Shape (Skewness and Kurtosis).

UNIT V: Sampling and Estimation (04 Hours)

Sampling Techniques (Simple Random, Stratified, Systematic, and Cluster Sampling), Sampling Distributions of Sample Mean and Sample Proportion, Central Limit Theorem, Point Estimation, Interval Estimation, and Confidence Intervals for Population Means and Proportions.

UNIT VI: Correlation, Regression, and Non-Parametric Methods (04 Hours)

Correlation Analysis (Pearson's and Spearman's Correlation Coefficients), Simple and Multiple Linear Regression, Model Evaluation and Diagnostics (Coefficient of Determination (R^2), Adjusted R^2, and Residual Analysis), Introduction to Non-Parametric Statistical Tests (Sign Test, Wilcoxon Rank-Sum Test, and Kruskal–Wallis Test), and Applications in Data Analysis**.**

List of Practical's: (Any Six)

1. Implement probability calculations using C/C++/Python for sample space, conditional probability, and Bayes' theorem.
2. Implement random variable and probability distribution functions (PMF, PDF, and CDF) using C/C++/Python.
3. Implement Binomial, Poisson, and Normal probability distributions using C/C++/Python.
4. Implement Uniform, Log-Normal, Beta, and Gamma probability distributions using C/C++/Python.
5. Implement measures of central tendency (Mean, Median, and Mode) using C/C++/Python.
6. Implement measures of dispersion (Range, Variance, Standard Deviation, and Coefficient of Variation) using C/C++/Python.
7. Implement frequency distribution and data visualization techniques using C/C++/Python.
8. Implement random, stratified, systematic, and cluster sampling techniques using C/C++/Python.
9. Implement confidence interval estimation for population mean and proportion using C/C++/Python.
10. Implement correlation, linear regression, and non-parametric statistical methods using C/C++/Python.

List of Course Projects:

1. Design and Implement a Student Performance Analysis System Using Statistical Techniques.
2. Design and Implement a Weather Data Analysis and Rainfall Prediction System Using Probability Distributions.
3. Design and Develop a Sales Forecasting System Using Regression Analysis.
4. Implement a Medical Data Analysis System for Disease Risk Prediction Using Statistical Methods.
5. Design and Implement a Traffic Flow Analysis and Prediction System Using Probability Models.
6. Design and Develop a Customer Survey Analysis System Using Descriptive Statistics.
7. Implement a Quality Control and Defect Analysis System for Manufacturing Processes.
8. Design and Implement a Stock Market Trend Analysis System Using Correlation and Regression.
9. Design and Develop a Crime Data Analysis and Visualization System Using Statistical Techniques.
10. Implement a Population Growth Analysis and Forecasting System Using Probability Distributions.
11. Design and Implement an Air Quality Analysis and Pollution Monitoring Dashboard Using Statistical Methods.
12. Design and Develop a Smart Agriculture Data Analysis System for Crop Yield Prediction.
13. Implement a Healthcare Data Visualization and Statistical Analysis System.
14. Design and Develop an Election Survey and Opinion Poll Analysis System Using Sampling Techniques.
15. Implement a Business Analytics Dashboard for Statistical Data Analysis and Decision Support.

Text Books: (As per IEEE format)

1. *D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, 6th ed. Hoboken, NJ: Wiley, 2014.*
2. *D. S. Moore and G. P. McCabe, Introduction to the Practice of Statistics, 9th ed. New York, NY: W.H. Freeman and Company, 2016.*
3. S. P. Gupta, Statistical Methods, 44th ed. New Delhi, India: Sultan Chand & Sons, 2017.
4. *S. C. Gupta and V. K. Kapoor, Fundamentals of Applied Statistics, 4th ed. New Delhi, India: Sultan Chand & Sons, 2020.*

Reference Books: (As per IEEE format)

1. *D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, Hoboken, NJ: Wiley, 2014.*
2. *2. S. C. Gupta and V. K. Kapoor, Fundamentals of Applied Statistics, New Delhi, India: Sultan Chand & Sons, 2020.*
3. *3. D. S. Moore, G. P. McCabe, and B. A. Craig, Introduction to the Practice of Statistics, New York, NY: W.H. Freeman and Company, 2016.*
4. *4. M. R. Spiegel and L. J. Stephens, Schaum's Outline of Statistics, New York, NY: McGraw-Hill, 2008.*
5. *5. T. H. Wonnacott and R. J. Wonnacott, Introductory Statistics for Business and Economics, Hoboken, NJ: Wiley, 1990.*
6. *6. W. Mendenhall, R. J. Beaver, and B. M. Beaver, Introduction to Probability and Statistics, Boston, MA: Brooks/Cole, 2013.*
7.

Course Outcomes:

The student will be able to –

1. Illustrate basics of probability and Bayes rule.
2. Analyze and solve problems involving random variables and compute mathematical expectations for both discrete and continuous distributions
3. Apply discrete and continuous probability distributions in analyzing the probability models arising in engineering field.
4. Apply basic statistical concepts, including data types, descriptive statistics, and graphical representation of data.
5. Apply appropriate sampling techniques and use statistical estimation methods to infer population parameters from sample data with confidence.
6. Interpret statistical results and relationships using correlation, regression, and appropriate statistical tools or software.

Moocs Links and additional reading material: (If any)

1. Probability and Statistics – NPTEL
   https://nptel.ac.in/courses/111104032
2. Introduction to Probability and Statistics – NPTEL
   https://onlinecourses.nptel.ac.in/
3. Introduction to Statistics – Khan Academy
   https://www.khanacademy.org/math/statistics-probability
4. Introduction to Probability and Statistics – MIT OpenCourseWare
   https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2014/
5. Statistics with Python Specialization – Coursera (University of Michigan)
   https://www.coursera.org/specializations/statistics-with-python
6. Statistics Learning Path – DataCamp

https://www.datacamp.com/

7. Statistics and Probability – edX
https://www.edx.org/learn/statistics

8. Statistics for Data Science – Google (Machine Learning Crash Course)
https://developers.google.com/machine-learning/crash-course/statistics

Future Courses Mapping:

1. Deep Learning
2. Data Mining
3. Cloud Computing
4. Artificial Intelligence

Job Mapping: (Optional)

1. Data Analyst
2. Business Analyst
3. Statistical Analyst
4. Software Engineer (Data-Driven Applications)

CO - PO Mapping:

	Program Outcomes (PO)											PSO	
CO/ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	1		1									1	
CO2	1	1	1									1	1
CO3	1	1	1									1	1
CO4	1	1	1									1	
CO5	1	1	1									1	
CO6	1	1										1	
Average	1	1	1									1	1

FF No: 654

MM0105: Augmented Reality and Virtual Reality

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20 Marks
Lectures :	2 Hrs/week	GD/PPT:20 Marks
Practical :	0 Hrs/week	HA: 20Marks
Tutorial :	1 Hrs/week	MSE: 40Marks

Course Objectives:

1. To Understand immersive technologies, their taxonomy, tools, applications, and distinguish AR, VR, and MR.
2. To Learn core concepts, hardware, and development lifecycle of Virtual Reality systems.
3. To gain practical knowledge of VR application development using tools like Unity, Unreal Engine, and Blender, including basic scripting and templates.
4. To comprehend Augmented Reality principles, types, and image processing techniques
5. To familiarize with AR development tools such as Vuforia and Unity, addressing challenges and real-world AR application deployment
6. To explore Mixed Reality technologies, their use cases, security challenges, and technical requirements for MR application development

Course Relevance: The course is offered in S.Y. B.Tech. AIDS Department**.**

SECTION-I

Unit 1: Introduction - (5 Hours)

Immersive Basics, taxonomy, methods and techniques, tools, introduction to Virtual Reality, Augmented Reality and Mixed Reality, spectrum, difference between AR/VR/MR, Applications of AR/VR/MR, Online and Offline tools for AR/VR/MR, Consequences of Excessive AR/VR/MR.

Unit 2: Virtual Reality - (5 Hours)

Basics of Virtual Reality, Types, Augmented Reality, Head Mounted Device (HMD), Types and working along with different applications, Collaborative VR, Key components and benefits, VR Application Development Cycle, Comparison with Software Development Life Cycle, Web VR, Mobile VR. Tools, Use cases, Evolution of VR technology, Tools and Technologies for VR app development.

Unit 3: Virtual Reality App Development - (4 Hours)

Introduction to Virtual Reality App development, Software's, Unity, Blender, Unreal Engine, Blueprint, Types, Templates in Unreal Engine development, Basics of Unity and, introduction to game development using unity.

SECTION-II

Unit 4: Augmented Reality - (5 Hours)

Introduction to Augmented Reality, types, mobile AR/ Web AR-headset AR, working principle, image processing principles in AR app, Basics of image processing, feature points detection, Keypoint detection, Keypoint description.

Unit 5: Augmented Reality App Development - (5 Hours)

Tools and technologies for AR app development, Vuforia, Vuforia engine, functions, Vuforia and Unity, challenges of AR app development, Applications of Augmented Reality in real world scenarios.

Unit 6: Mixed Reality - (4 Hours)

Mixed Reality basics, Technological requirement in development and deployment of MR apps, Security issues with MR apps. Mixed Reality applications. Use case of mixed reality.

List of Tutorials:

Tutorial 1: To study and prepare a report on **Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR)** through a hands-on demonstration using the **ARLOOPA** application.
Tutorial 2: To create a **Unity** project in which a 3D cube moves using the **Arrow Keys** or **WASD** keyboard controls.

Tutorial 3: To implement a **Camera Follow** system in Unity that enables the camera to smoothly track the movement of a 3D player object.

Tutorial 4: To design and develop a basic **Virtual Reality (VR)** scene using the **A-Frame** framework, demonstrating fundamental VR concepts and interactions.

Tutorial 5: To create a **VR-like interactive environment** in Unity where the player can navigate using the keyboard and mouse and perform object interactions such as picking up and dropping objects without requiring a VR headset.

Tutorial 6: To create a **bouncing ball animation** in **Blender** using keyframes and the timeline, demonstrating fundamental animation principles such as **timing, squash and stretch, anticipation, and easing** to simulate realistic motion.

Tutorial 7: To develop an interactive AR application in Unity using Vuforia Engine that recognizes an image target and displays a 3D object on it, demonstrating the fundamentals of marker-based Augmented Reality.

Tutorial 8: To create a simple first-person 3D game in Unity by designing a basic environment with obstacles, implementing collision detection, score collection, and a game-over condition using C# scripting.

List of Course Projects:

21. Virtual Campus Tour
22. AR-Based Furniture Placement
23. 3D Maze Game
24. VR Escape Room
25. Virtual Museum
26. Virtual Art Gallery
27. AR Solar System
28. AR Animal Learning App
29. Virtual Classroom
30. VR House Walkthrough
31. First-Person Adventure Game
32. 12 Endless Runner Game
33. Treasure Hunt Game
34. Car Parking Simulator
35. Traffic Signal Simulation
36. Virtual Shopping Mall
37. AR Business Card
38. Interactive Science Laboratory
39. Virtual Zoo
40. 3D Puzzle Game

List of Course Seminar Topics:

21. To study the fundamentals of Augmented Reality (AR) and its real-world applications.
22. To explore the concepts, components, and applications of Virtual Reality (VR).
23. To understand the principles of Mixed Reality (MR) and its role in immersive computing.
24. To study Extended Reality (XR) and its impact on the future of digital interaction.
25. To explore the architecture, features, and applications of the Metaverse.
26. To study the use of Unity Engine for developing interactive AR and VR applications.
27. To understand the role of Blender in 3D modeling, animation, and game asset creation.
28. To explore the features and applications of the A-Frame framework for WebVR development.
29. To study the working principles of marker-based and markerless Augmented Reality systems.
30. To understand the applications of AR and VR technologies in the healthcare sector.
31. To study the use of AR and VR in education and immersive learning environments.
32. To explore the applications of Virtual Reality in the gaming and entertainment industry.
33. To understand the role of Digital Twin technology in smart industries and manufacturing.
34. To study the applications of Spatial Computing in next-generation computing systems.
35. To explore Human-Computer Interaction (HCI) techniques in AR and VR environments.
36. To understand gesture recognition and motion tracking technologies used in immersive systems.
37. To study the security, privacy, and ethical challenges associated with AR and VR technologies.
38. To explore the integration of Artificial Intelligence (AI) with AR and VR applications.
39. To study the latest trends and future developments in AR, VR, MR, and XR technologies.
40. To explore career opportunities and emerging research areas in AR, VR, XR, and game development.

Text Books: (As per IEEE format)

1. ***Tony Parisi , Learning Virtual Reality:*** *Developing Immersive Experiences and Applications for Desktop, Web, and Mobile, Wiley, 2015.*

2. ***Murray Ramirez, Virtual Reality for Beginners!:*** *How to Understand, Use & Create with VR, by, 2016.*

3. ***Roger Froze, Augmented Reality For Beginners!:*** *Principles & Practices for Augmented Reality & Virtual Computers, 2016*

Reference Books and E-Resources

3. *Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, 3D User Interfaces, Theory and Practice, Addison Wesley, USA, 2005.*
4. *Oliver Bimber and Ramesh Raskar, Spatial Augmented Reality: Merging Real and Virtual Worlds, 2005.*
Example - H. Schmidt-Walter, R. Kories; 'Electrical Engineering. A Pocket Reference'; Artech House, 2007. Accessed: Oct. 16, 2016. [Online]. Available: https://ebookcentral.proquest.com

Course Outcomes:

7. Describe the key concepts and applications AR/VR/MR.
8. Apply the AR/VR application development life cycle using different tools.
9. Develop VR applications using suitable VR tools.

10. Understand Augmented Reality and image processing.
11. Develop AR applications using suitable AR tools.
12. Understand advances in Mixed Reality.

For MOOCs and other learning Resource

4. https://www.deepar.ai/demos
5. https://app.vectary.com/
6. https://builtin.com/articles/what-is-webar

Future Courses Mapping:

7. Provides the basic knowledge of AR, VR, and MR technologies.
8. Helps students develop AR and VR applications using modern tools.
9. Builds a foundation for game development and 3D visualization.
10. Prepares students for advanced courses in Extended Reality (XR) and the Metaverse.
11. Enhances practical skills in Unity, Blender, and AR development.
12. Supports career opportunities in AR/VR development, simulation, and immersive technologies.

Job Mapping:

This course equips students with the knowledge and practical skills required for careers in AR/VR, Unity, and immersive application development. It prepares students for entry-level roles such as AR Developer, VR Developer, Unity Developer, and Game Developer.

CO - PO Mapping:

	Program Outcomes (PO)											PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	1										1		
CO2			2										
CO3			3		2	1						2	3
CO4	1												
CO5			3		2	1						2	3
CO6								2					
Average	1		2.6		2	1		2			1	2	3

FF No: 654

MM0106: Image Processing

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20 Marks
Lectures :	2 Hrs/week	GD/PPT:20 Marks
Practical :	0 Hrs/week	HA: 20Marks
Tutorial :	1 Hrs/week	MSE: 40Marks

Course Prerequisites:

1. Basic knowledge of computer programming (Python/MATLAB/C preferred).
2. Fundamental understanding of mathematics, including matrices and vectors.
3. Basic concepts of computer graphics and digital images.

Course Objectives:

1. To understand the basics of digital images, color models, and image formats.
2. To apply enhancement techniques in spatial and frequency domains.
3. To learn segmentation and feature extraction methods.
4. To study image compression techniques and their applications.
5. To use image transforms for image analysis and processing.
6. To apply image processing methods to real-time problem solving.

Course Relevance:

Provides the fundamentals of digital image processing for applications in AI, Computer Vision, Medical Imaging, and Robotics.

SECTION-I

Unit 1: Introduction to Image processing (5 Hours)

Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YC_bC_r and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF,png, raster image format). Image sampling and quantization.

Unit 2: Image Enhancement (5 Hours)

Thresholding (Global, Adaptive, and Otsu's Method), Spatial Domain Techniques including Image Negative, Contrast Stretching, Histogram Processing and Histogram Equalization, Image Subtraction, Image Smoothing using Mean, Gaussian, and Median Filters, Image Sharpening using High-Pass, Laplacian, and Sobel Filters, and Frequency Domain Techniques including Butterworth Low-Pass Filter and High-Pass Filter.

Unit 3: Image Analysis (5 Hours)

Image Segmentation: Threshold-based, Edge-based, Region-based, Watershed, and Clustering-based Segmentation; Edge Detection using Sobel, Prewitt, Canny, and Laplacian Operators. Feature Extraction: Boundary Representation (Chain Code, Fourier Descriptors), Region Properties (Area, Perimeter, Euler Number, Eccentricity, Moments), Texture and Color Features, and Introduction to CNN-based Feature Extraction and Image Segmentation.

SECTION-II

Unit 4: Image Compression (4 Hours)

Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), scalar and vector quantization

Unit 5: Object recognition (4 Hours)

Need of Object Recognition, Automated Object Recognition System, Pattern and Pattern Class, Relationship between Image Processing and Object Recognition, Approaches to Object Recognition: Template Matching, Feature-Based Recognitio, Haar Cascade Classifier.

Unit 6: Image Transform (5 Hours)

Introduction to two dimensional orthogonal and unitary transforms, properties of nitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine,Discrete Wavelet Transform (DWT),Walsh–Hadamard Transform (WHT) , Applications of transforms in Image processing.

List of Practical's: (Any ….)

List of Experiments

1. Write matlab code to display following binary images
Square, Triangle , Circle

- Write matlab code to perform following operations on images
- Flip Image along horizontal and vertical direction.
- Enhance quality of a given image by changing brightness of image.
- Image negation operation.
- Change contrast of a given Image.

2. Write Matlab code to implement pseudo colouring operation of a given image.

Write Matlab Code for Pseudo Colour of Image by using Gray to colour transform.

3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.

4. Write matlab code to find following statistical properties of an image.

- Mean
- Median
- Variance
- Standard deviation
- Covariance.

5. Write matlab code to enhance image quality by using following techniques

- Logarithmic transformation
- Histogram Equalization
- Gray level slicing with and without background.
- Inverse transformation.

6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.

7. Write matlab code to enhance image quality by using following techniques

- Low pass and weighted low pass filter.
- Median filter.
- Laplacian mask.

8. Write matlab code for edge detection using Sobel, Prewitt and Roberts operators.

9. Write C-language code to find out Huffman code for the following word COMMITTEE.

10. Write matlab code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3,Y- 0.1).

11. Write matlab code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.

12. Develop an AI-inspired image stylization and enhancement system using classical image processing techniques.(GenAI based)

List of Tutorials:

1) Introduction to image processing tools and environment setup using Python or MATLAB.
2) Reading, displaying, and basic manipulation of images including color space conversions.
3) Image sampling and quantization with analysis of resolution and bit depth effects.
4) Spatial domain enhancement techniques like contrast stretching and histogram equalization.
5) Fourier Transform and frequency domain filtering for image analysis.

6) Noise addition and image restoration using mean, median, and inverse filtering.
7) Edge detection and morphological operations such as dilation and erosion.
8) Image compression techniques and color-based segmentation using the HSI model.

List of Course Projects:

1. Real-Time Multi-Object Detection and Tracking using YOLO and OpenCV
2. Automatic Face Recognition with Face Mask Detection using Deep Learning
3. Vision-Based Driver Drowsiness and Distraction Detection System
4. Deep Learning-Based Medical Image Segmentation for Tumor Detection
5. AI-Based Automatic Number Plate Recognition (ANPR) System
6. Image Forgery and Deepfake Detection using CNN and Image Forensics
7. Plant Leaf Disease Detection and Severity Analysis using Deep Learning
8. Crack Detection in Civil Infrastructure using CNN and Image Processing
9. Satellite Image Land Cover Classification using Deep Learning
10. Human Pose Estimation and Activity Recognition using Computer Vision
11. OCR-Based Intelligent Document Analysis and Information Extraction
12. Vision-Based Smart Attendance System using Face Recognition and Liveness Detection

List of Course Seminar Topics: (*if applicable to your course*)

1. Human Visual System (HVS) and Color Models in Digital Image Processing
2. Image Sampling and Quantization: Concepts and Applications
3. Image Enhancement Techniques in Spatial and Frequency Domains
4. Histogram Equalization and Contrast Enhancement Methods
5. Image Filtering Techniques for Noise Removal
6. Edge Detection Techniques: Sobel, Prewitt, Canny, and Laplacian
7. Image Segmentation Techniques and Their Applications
8. Feature Extraction Methods in Digital Images
9. Image Compression Standards: JPEG and Lossless Compression Techniques
10. Haar Features and Haar Cascade Classifier for Object Detection
11. Object Recognition Approaches: Template Matching and Feature-Based Recognition
12. Two-Dimensional Discrete Fourier Transform (DFT) in Image Processing
13. Discrete Wavelet Transform (DWT) for Image Analysis and Compression
14. Applications of Image Processing in Medical Imaging, Remote Sensing, and Industrial Inspection
15. AI-Assisted Image Processing: Integration of Classical Image Processing with Deep Learning

Text Books: (As per IEEE format)

1. Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition,2010.
2. Anil K. Jain, 'Fundamentals of Digital Image Processing', Pearson, 2002

Reference Books: (As per IEEE format)

1. Kenneth R. Castleman, 'Digital Image Processing', Pearson, 2006.
2. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, 'Digital Image Processing using MATLAB', Pearson
Education, Inc., 2011.
3. D,E. Dudgeon and RM. Mersereau, 'Multidimensional Digital Signal Processing', Prentice Hall Professional
Technical Reference, 1990
4. William K. Pratt, 'Digital Image Processing', John Wiley, New York, 2002
5. Milan Sonka et al 'Image processing, analysis and machine vision', Brookes/Cole, Vikas Publishing House, 2nd
edition, 1999.

Course Outcomes:

The student will be able to –

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

1. Describe various image models and their properties.
2. Apply spatial filtering techniques for image enhancement.
3. Identify and implement image segmentation techniques.
4. Apply appropriate lossless and lossy compression methods.
5. Implement basic object recognition techniques.
6. Use image transforms to analyze and process digital images.

Moocs Links and additional reading material: (If any)

Future Courses Mapping:

Computer Vision, Deep Learning, Pattern Recognition, and Artificial Intelligence.

Job Mapping: (Optional)

1. Image Processing Engineer
2. Computer Vision Engineer
3. AI/ML Engineer
4. Data Scientist
5. Robotics Engineer
6. Machine Vision Engineer

CO - PO Mapping:

	ProgramOutcomes(PO)PSO												
CO/PO	PO1	PO 2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	1	1	2	-	1	-	-	-	-	-	-	3	1
CO2	2	1	2	2	2	-	-	-	-	-	-	3	2
CO3	3	2	2	2	2	-	-	-	-	-	-	3	2
CO4	3	2	2	2	2	-	-	-	-	-	-	2	3
CO5	2	2	2	2	2	-	-	-	-	-	-	2	3
CO6	3	3	2	2	2	-	-	-	-	-	-	3	2
Average	**2.33**	**1.83**	**2**	**1.67**	**1.83**							**2.67**	**2.167**

FF No.: 654

MM 0107: Computer Graphics

Teaching Scheme		Examination Scheme
Credits:	3	CP: 20 Marks
Lectures :	2 Hrs/week	GD/PPT:20 Marks
Practical :	0 Hrs/week	HA: 20Marks
Tutorial :	1 Hrs/week	MSE: 40Marks

Course Objectives:

1. To understand the fundamentals and applications of computer graphics.
2. To study graphics hardware and the graphics pipeline.
3. To implement algorithms for drawing and transforming 2D/3D objects.
4. To explore concepts of projections, shading, and visibility.
5. To introduce animation techniques and graphics programming tools.
6. Describe the fundamentals of animations and gaming.

Course Relevance: The course is offered in S.Y. B.Tech. AIDS Department.

SECTION-I

Unit 1: Introduction **(5 Hours)**

Introduction to Computer Graphics- Definition, history and scope of computer graphics, pixel, Pixel dimensions,. Typical graphics pipeline (CPU → GPU), Graphics file formats – raster (BMP, PNG, JPEG, GIF, TIFF), vector (SVG), 3-D (OBJ, FBX, glTF), compression basics, Framebuffer concept, Color Models- RGB, CMYK, HSL,HSV Raster-scan vs. random-scan systems

Unit 2: Output Primitives & Attributes- **(5 Hours)**

Scan-conversion of basic primitives- DDA and Bresenham line algorithms (integer & floating-point forms), Mid-point circle, Polygon rasterisation – scan-line algorithm, inside-outside tests, edge tables, Area-fill algorithms – boundary-fill, flood-fill, seedfill variants, Primitive attributes –intensity mapping, point/line styles, area patterns, Anti-aliasing – super-sampling, area sampling,

Unit 3: 2-D Geometric Transformations & Viewing- **(5 Hours)**

Homogeneous coordinates 3 × 3 transformation matrices, Elementary transforms – translation, rotation, uniform & differential scaling, reflection, shear, Composite transforms; rotation about an arbitrary pivot, Window-to-viewport mapping; aspect ratio preservation, 2-D clipping, Line clipping – Cohen-Sutherland, , Polygon clipping.

SECTION-II

Unit 4: 3D Graphics & Projections- (5 Hours)

3-D Cartesian & homogeneous coordinate systems, 3-D object representations – polygon meshes, quad-meshes, boundary-rep, constructive solid geometry, parametric curves & surfaces (brief), 4 × 4 transformation matrices – translation, rotation about coordinate & arbitrary axes, scaling, reflection, Introduction to Parallel projections – orthographic, axonometric (isometric, dimetric, trimetric), oblique.

Unit 5: Illumination, Shading & Visible-Surface Algorithms- (4 Hours)

Light–material interaction – ambient, diffuse (Lambert), specular (Phong), Shading methods – flat, Gouraud, Phong; normal interpolation, per-fragment shading, texture mapping overview, Shadow generation basics – shadow mapping

Unit 6: Animation & Advanced Applications- (4 Hours)

Principles of animation – key-framing, forward & inverse kinematics Motion capture pipeline, basics of skeletal animation, Morphing techniques – linear blend-shape

List of Practical's:

1. Pixel Plotting & Line Drawing Implement DDA and Bresenham's line drawing algorithms. Compare outputs and discuss accuracy and efficiency.
2. Circle and Ellipse Drawing Implement midpoint algorithms for drawing circles and ellipses using integer arithmetic.
3. Polygon Drawing and Filling Draw arbitrary polygons and implement area-filling algorithms (boundary-fill and flood-fill).
4. 2D Transformations Apply translation, rotation, scaling, and reflection to 2D objects using transformation matrices.
5. Polygon Clipping Implement Sutherland-Hodgman polygon clipping and Cohen-Sutherland line clipping algorithms.
6. Viewport Transformation Demonstrate window-to-viewport mapping with normalized coordinates.
7. 3D Transformations Apply 3D transformations (translation, rotation, scaling) to basic models. Use GenAI (eg. ShapeNet, DreamFusion) to generate the 3D model that can be transformed using learned techniques
8. Projection Techniques Implement orthographic and perspective projections for simple 3D objects.
9. Lighting and Shading Implement flat, Gouraud, and Phong shading models on 3D surfaces.
10. Basic Animation and Object Export Create a key-frame animation sequence and export a 3D object in OBJ format.

List of Tutorials:

Tutorial 1: Understanding Graphics Pipeline and Color Models: Fundamentals of Computer Graphics: Pipeline, Formats, and Color Models (Presentation)

Tutorial 2: Line and Circle Drawing Algorithms

- Draw a line from (2, 2) to (10, 6) using Bresenham's algorithm.
- Use the Midpoint Circle algorithm to draw a circle with radius = 5 centered at (0,0).

Tutorial 3: 2D Geometric Transformations & Viewing

- Translate a triangle with vertices (1,2), (3,5), (5,1) by (2,3).
- Rotate a square (0,0), (0,2), (2,2), (2,0) by 90° about origin.
- Apply Cohen-Sutherland algorithm to clip lines from (-5,5) to (5,7) in the window (- 4,-4) to (4,4).

Tutorial 4: 3D Graphics & Projections

- Rotate a point (1,0,0) about Z-axis by 90°.
- Apply composite transformation (translate → scale → rotate) on a 3D object.

Tutorial 5: Illumination, Shading & Visible Surface Algorithms (Writeup)

- Explain differences using triangular polygon and normal vectors.
- Compare flat, Gouraud, and Phong shading techniques.
- Explain the z-buffer and painter's algorithm.

Tutorial 6: Animation & Advanced Applications

- Animation Techniques and Modern Applications in Computer Graphics (Presentation)

List of Course Projects:

21. Interactive Computer Graphics Learning Toolkit
22. 2D Graphics Editor (Mini Paint Application)
23. Graphics Algorithm Visualizer
24. Raster Graphics Simulator
25. Simple CAD Drawing System
26. Image Format and Compression Explorer
27. 2D Game Graphics Engine
28. Interactive Polygon Editor
29. Digital Map Viewer
30. 3D Projection Visualizer
31. Computer Pipeline simulator
32. 3D Wireframe Modeling System
33. Virtual Architecture Viewer
34. Interactive Lighting and Shading Demonstrator
35. 3D Material and Texture Viewer

36. Texture Mapping Simulator
37. 3D Animated Robot Arm
38. Motion Capture Visualization System
39. 3D Car Driving Simulator
40. Virtual Museum Tour

List of Course Seminar Topics:

19. **Learning tools, visualization of graphics concepts, interactive simulations, educational applications.**
20. **3D Graphics Pipeline and Projection Techniques 3D:** graphics pipeline, homogeneous coordinates, orthographic, axonometric, and oblique projections
21. **Computer-Aided Design (CAD):** Principles and Modern Applications CAD systems, geometric modeling, 3D object creation, engineering applications
22. **3D Object Representation Techniques in Computer Graphics** Polygon meshes, quad meshes, boundary representation (B-Rep), constructive solid geometry (CSG)
23. **Geometric Transformations in 3D Computer Graphics** Translation, rotation, scaling, reflection, arbitrary axis rotation, transformation matrices
24. **Illumination Models in Computer Graphics** Ambient, diffuse (Lambert), specular (Phong), light-material interaction
25. **Shading Techniques for Real-Time Rendering** Flat, Gouraud, and Phong shading, normal interpolation, per-fragment shading
26. **Texture Mapping Techniques and Material Design in 3D Graphics** UV mapping, texture coordinates, procedural textures, material properties
27. **Shadow Mapping Algorithms for Realistic Rendering** Shadow generation, depth maps, shadow mapping pipeline, optimization techniques
28. **Dynamic Lighting in Virtual Environments** Multiple light sources, real-time lighting, global vs. local illumination
29. **Building Virtual Art Galleries Using Modern Graphics Techniques** 3D scene creation, lighting, camera navigation, interactive virtual environments
30. **Animation Principles in Computer Graphics** Key-framing, timing, squash and stretch, anticipation, follow-through
31. **Forward and Inverse Kinematics in Character Animation** Skeletal animation, joint hierarchy, robotic and gaming applications
32. **Motion Capture Technology:** Motion capture pipeline, sensors, data processing, animation retargeting
33. **Skeletal Animation Techniques for Modern Games** Bone hierarchy, skinning, rigging, animation blending
34. **Morphing Techniques in Computer Graphics** Blend shapes, facial animation, image morphing, character expression
35. **Applications of 3D Computer Graphics in Industry** Gaming, films, architecture, healthcare, education, virtual reality
36. **Future Trends in Computer Graphics:** AR, VR, AI, and Digital Twins Emerging technologies,

immersive environments, AI-assisted graphics, Industry 4.0

Text Books: (As per IEEE format)

4. *Donald Hearn & M. Pauline Baker, "Computer Graphics with OpenGL", 4th Edition, Pearson, 2010.*
5. *Edward Angel & Dave Shreiner, "Interactive Computer Graphics: A Top-Down Approach with WebGL", 8th Edition, Pearson, 2020.*
6. *James D. Foley , A. van Dam, John F. Hughes,, et al. "Computer Graphics: Principles and Practice", 3rd Edition, Addison-Wesley, 2019.*

Reference Books: (As per IEEE format)

5. *F. S. Hill Jr. & Stephen Kelley, "Computer Graphics Using OpenGL", 3 rd Edition, Prentice Hall, 2006.*
6. *David F. Rogers & J. Alan Adams, "Mathematical Elements for Computer Graphics", 2nd Edition, McGraw-Hill, 1990.*
7. *Steve Marschner & Peter Shirley (eds.), "Fundamentals of Computer Graphics", 5th Edition, CRC Press, 2021.*
8. *Steven Harrington. Computer Graphics, "A Programming Approach", 2nd Edition, McGraw-Hill,*

Course Outcomes:

The student will be able to –

Upon completion of the course, students will be able to –

1. Explain the architecture, hardware, and applications of computer graphics system
2. Implement raster algorithms for drawing primitives with color and antialiasing.
3. Apply 2D transformations and clipping using matrix operations.
4. Manipulate 3D models and projections for viewing and rendering.
5. Apply shading and visibility techniques to enhance realism.
6. Create simple animations and explore graphics file formats and shaders.

Moocs Links and additional reading material:

https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs90

Future Courses Mapping:

7. Understand Graphics Pipeline & GPU Programming.
8. Implement and simulation of primitives of computer graphics.
9. Geometric Modeling.
10. Projection & Viewing .
11. Illumination & Shading with Real-Time Rendering, Game engines, visualization.
12. Understand Animation Game Development.

Job Mapping:

Graphics Software Engineer, Game Developer, AR/VR Developer, Computer Vision Engineer, UI/UX Designer, CAD Engineer, Animation Programmer, Digital Twin Developer, Metaverse Developer, AI Graphics Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**		**PSO1**	**PSO2**
CO1	2	1	1	-	-	-	-	-	-	-	-		1	1
CO2	1	2	-	-	-	-	-	-	-	-	-		1	1
CO3	2	1	1	-	-	-	-	-	-	-	-		-	1
CO4	-	1	1	-	-	-	-	-	-	-	-		1	1
CO5	2	2	1	-	-	-	-	-	-	-	-		1	1
CO6	2	1	2		3	-	-	-	-	-	-		2	3
Average	1.5	1.3	1	-	0.5	-	-	-	-	-	-		1	1.3

FF No.: 654

HS2003: From Campus To Corporate – II

Teaching Scheme		Examination Scheme
Credits:	2	CP: NA
Lectures :	2 Hrs/week	CVV:NA
Practical :	0 Hrs/week	MSE(R): 50 Marks
Tutorial :	0 Hrs/week	ESE(R) : 50 Marks

Introduction to the Corporate World Understanding organizational structure and hierarchy, Work culture differences: campus vs. corporate,

Employer expectations from fresh graduates, Time management and ownership in corporate settings

Professional Communication Skills: Verbal and non-verbal communication, Email and business writing etiquette, Presentation skills and use of visual aids, Listening skills and telephone etiquette,

Soft Skills and Interpersonal Effectiveness: Body language, grooming, and first impressions, Conflict resolution and negotiation skills, Team dynamics and collaboration, Assertiveness vs. aggressiveness

Resume Building and Job Preparation: Building an effective resume and cover letter, Identifying strengths and achievements, Preparing for technical and HR interviews, Handling rejections and feedback

Group Discussions and Personal Interviews: Group discussion formats and evaluation criteria, Strategies for initiating, contributing, and summarizing, Mock interviews with feedback, STAR technique for answering behavioral questions,

Corporate Etiquette and Workplace Ethics: Meeting and greeting protocol, Dining and social etiquette, Work ethics, punctuality, confidentiality, Respect for diversity and inclusion in the workplace

Adaptability and Emotional Intelligence: Handling pressure, deadlines, and ambiguity, Self- awareness and emotional regulation, Empathy and workplace relationships, Managing feedback and continuous learning,

Introduction to Project Management Basics: Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

Faculty are supposed to do conduct following in the class

- Resume and LinkedIn profile workshops
- Mock interviews and GD sessions
- Role plays: workplace scenarios, conflict handling
- Business email writing exercises
- Presentation and elevator pitch sessions

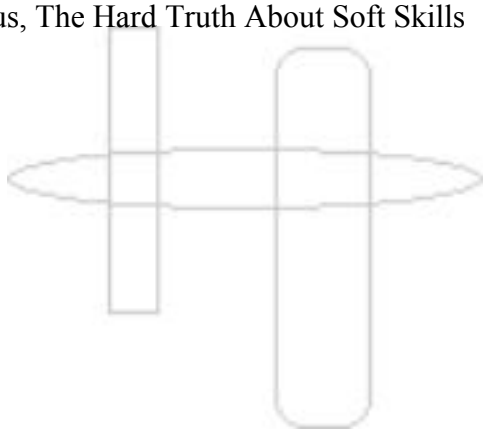
Books:

1. Dale Carnegie, How to Win Friends and Influence People

2. Stephen R. Covey, 7 Habits of Highly Effective People

3. Shital Kakkar Mehra, Business Etiquette: A Guide for the Indian Professional

4. Peggy Klaus, The Hard Truth About Soft Skills



FF No.: 654

HS2004: Reasoning & Aptitude Development-4

Teaching Scheme	Examination Scheme
Credits : 2	CP: -- Marks
Lectures : -- Hrs/week	GD/PPT/HA: -- Marks
Practical : -- Hrs/week	MSE (W/O): -- Marks
Tutorial : 1 Hrs/week	ESE (W/O): 100 Marks

Prerequisites:

- Basic knowledge of English grammar and vocabulary.
- Fundamental understanding of sentence construction and communication skills.
- Basic arithmetic and mathematical concepts (Class XII level).
- Elementary logical and analytical reasoning skills.
- Familiarity with problem-solving techniques.
- Basic computer literacy and programming concepts.
- Exposure to at least one programming language (C/C++/Java/Python) is desirable.

Course Objectives:

- Develop proficiency in English grammar, vocabulary, reading comprehension, and effective written communication.
- Strengthen logical, analytical, critical, and reasoning abilities to solve diverse aptitude-based problems.
- Apply quantitative and mathematical concepts to solve real-world numerical and analytical problems efficiently.
- Interpret, analyze, and synthesize information from textual, graphical, tabular, and numerical data for informed decision-making.
- Develop computational thinking and programming aptitude through logic building, pseudocode, algorithmic problem solving, and debugging techniques.
- Enhance coding proficiency, optimization skills, and problem-solving abilities to perform effectively

in programming assessments, campus recruitment tests, and technical interviews.

Course Outcomes:

After completion of the course, student will be able to

1. **Apply** appropriate English language skills, including grammar, vocabulary, reading comprehension, and effective communication techniques to improve verbal and written proficiency.
2. **Analyze** logical reasoning problems, critical thinking scenarios, and data interpretation tasks to arrive at accurate and evidence-based conclusions.
3. **Solve** quantitative aptitude problems using mathematical concepts, algebraic reasoning, probability, geometry, and analytical techniques in real-life contexts.
4. **Develop** efficient algorithmic solutions by applying computational thinking, programming fundamentals, debugging, optimization techniques, and time complexity analysis to solve coding problems.

SECTION1
English Language and Communication Skills: English tenses (Present, Past, Future), active and passive voice, sentence correction, parts of speech, syntax, sentence structure, subject–verb agreement, reading comprehension, skimming and scanning, inference, tone, theme, intent identification, contextual vocabulary. **Logical Reasoning and Analytical Thinking:** Deductive reasoning (syllogisms, coding–decoding, statement–conclusion, data sufficiency, directional sense, logical sequencing), inductive reasoning (analogies, classification, number series, sequence completion, trend identification), abductive and critical reasoning (assumptions, arguments, cause–effect, logical evaluation, decision making), information gathering, interpretation and synthesis using graphs, charts, tables and multi-source data.
SECTION2
Quantitative Aptitude and Mathematical Reasoning: Number systems, integers, fractions, decimals, divisibility, HCF and LCM, prime and rational numbers, algebraic expressions, linear equations and inequalities, proportional reasoning, speed–time–distance, profit and loss, percentages, averages, ages, mixtures, permutations and combinations, probability, geometry, spatial reasoning, pattern recognition and quantitative problem solving. **Programming Aptitude and Automata Skills:** Logic building, pseudocode, programming fundamentals, basic coding problems, program correctness, execution efficiency, intermediate coding challenges, data structures, algorithms, debugging, optimization techniques, time complexity analysis, industry best coding practices, Automata Pro IT, Automata Pro Max and comprehensive mock

assessments.	
Text Books: ***(As per IEEE format)***	
1	R. S. Aggarwal, *Quantitative Aptitude for Competitive Examinations*, Revised and Enlarged Edition. New Delhi, India: S. Chand and Company Ltd., 2017.
2	R. S. Aggarwal, *A Modern Approach to Verbal & Non-Verbal Reasoning*. New Delhi, India: S. Chand Publishing.
3	Dr. M. B. Lal and A. Gupta, *CSAT Logical Reasoning & Analytical Ability*. Agra, India: Upkar Prakashan.
4	G. Laakmann McDowell, *Cracking the Coding Interview: 189 Programming Questions and Solutions*, 6th ed. Palo Alto, CA, USA: CareerCup, 2015.
Reference Books: ***(As per IEEE format)***	
1	B. H. Dowden, *Logical Reasoning*. California State University, Sacramento, CA, USA, 2011.
2	*501 Challenging Logic and Reasoning Problems*, 2nd ed. New York, NY, USA: LearningExpress, LLC, 2005.
3	*Logic and Reasoning Problems*. LearningExpress Practice Series.

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)												
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO 1**	**PSO 2**
CO1	–	–	–	–	–	–	–	–	3	–	2		
CO2	2	2	–	–	–	–	–	–	1	–	2		
CO3	2	2	–	–	–	–	–	–	1	1	2	2	
CO4	2	2	3	3	3	–	1	–	1	1	3		3
Avg	**2**	**2**	**3**	**3**	**3**	**–**	**1**	**–**	**1.5**	**1**	**2.25**	2	3

FF No.: 654

AI2311: Design Thinking IV

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	0 Hrs/week	ESE: 100 Marks
Tutorial :	1 Hrs/week	

Course Prerequisites:

Basic knowledge of research work, research paper and patent.

Course Objectives:

1. Understand the concepts of design thinking approaches
2. Apply both critical thinking and design thinking in parallel to solve problems
3. Apply some design thinking concepts to their daily work
4. To provide ecosystem for students and faculty for paper publication and patent filing

Course Relevance:

The course is offered in S.Y. B.Tech. to all branches of Engineering

SECTION-I

Contents for Design Thinking :

- Structure of The paper Journal List (Top 50 Journals)
- Selection of the journal
- Use of various online journal selection tools
- Plagiarism checking
- Improving contents of the paper
- Patent drafting
- Patent search Filing of patent Writing answers to reviewer questions
- Modification in manuscript Checking of publication draft

Assessment Scheme:
Publication of paper or patent

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Understand the importance of doing Research

CO2: Interpret and distinguish different fundamental terms related to Research

CO3: Apply the methodology of doing research and mode of its publication

CO4: Write a Research Paper based on project work

CO5: Understand Intellectual property rights

CO6: Use the concepts of Ethics in Research

CO-PO Mapping:

	Program Outcomes (POs)												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2	1	2	2	1	3	1	3	1	1	2	2	2
CO2	1	2	1	1	0	1	1	3	1	2	0	3	1
CO3	2	2	2	2	2	2	2	3	2	1	2	3	2
CO4	2	2	1	1	2	2	1	3	1	2	1	3	2
CO5	2	2	2	2	2	2	2	3	1	2	1	2	2
CO6	2	2	2	2	2	2	2	3	2	0	2	2	2
Average	2	2	1	2	2	2	2	3	2	2	2	3	2

FF No. : 654

AI 2312: Engineering Design & Innovation IV

Teaching Scheme		Examination Scheme
Credits:	2	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	4 Hrs/week	MSE : 30 Marks
Tutorial :	0 Hrs/week	ESE : 70 Marks

Course Prerequisites:

Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, the curriculum can be enriched with semester-long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy, and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question challenge, or problem to research and respond to and/or solve. It brings students not only to know,

understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

SECTION-I

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Text Books: (As per IEEE format)

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017
2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.
3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro

Reference Books: (As per IEEE format)

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.
2. Project management core textbook, second edition, Indian Edition , by Gopalan.
3. The Art of Agile Development. By James Shore & Shane Warden.

Moocs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

Upon completion of the course, student will be able to –

On completion of the course, learner will be able to–

CO1: Identify the real life problem from a societal need point of view
CO2: Choose and compare alternative approaches to select the most feasible one
CO3: Analyse and synthesize the identified problem from a technological perspective
CO4: Select the best possible solution to solve the problem.
CO5: Design & Develop a working model of the proposed solution.
CO6: Testing and validating product performance

Future Courses Mapping:
Major Project

Job Mapping:

Job opportunities that one can get after learning this course
Software Engineer. Software Developer, IT Engineer, Research Associate.

CO - PO Mapping:

	Program Outcomes (POs)												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	3	2			3				2	2	2	3
CO2				3	2	2		2				2	
CO3	2		3		3			2	2			2	2
CO4		2				3						2	
CO5			3				2		2				
CO6		2			3				2	3			
Average	0.83	1.16	1.33	0.5	1.33	1.33	0.33	0.66	0.5	0.83	0.33	1.33	0.83

TY . B. Tech. Artificial Intelligence and Data Science
AY 2026-27

FFNo.:654

AI3201: Artificial Intelligence

Teaching Scheme	Examination Scheme
Credits: 4	ESE:40 Marks
Lectures : 3 Hrs/week	PRACT+CVV: 60 Marks
Practical : 2 Hrs/week	
Tutorial : 0 Hrs/week	

Course Prerequisites:

Computer Programming and Data Structures

Mathematical Foundations of Computer Science

Linear algebra, data structures and algorithms, and probability

Course Objectives:

CO1 (Unit 1): To learn the distinction between optimal reasoning Vs. human like reasoning

CO2 (Unit 2): To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.

CO3 (Unit 3): Implement optimization models of computation and processing in real world application of intelligent agents

CO4 (Unit 4): To understand the applications of AI, namely game playing, theorem proving, and machine learning.

Course Relevance:

Technologies driven by artificial intelligence (AI) have transformed industries and everyday life. The possibilities for AI applications are virtually unlimited and sought after in practically every industry segment. That's why global organizations are actively recruiting professionals with specialized skills and proficiencies needed to develop future AI technological innovations.

SECTION-I

Unit 1 Fundamentals of Artificial Intelligence **(7 hours)**

Introduction: A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, Types of production systems, Turing Test. Intelligent Agents: Agents and Environments, concept of rationality, the nature of

environments, structure of agents, problem solving agents, problem formulation. Formulation of problems: Vacuum world, 8 queens, Route finding, robot navigation. [CO1, CO2] [PO1, PO2]

Unit 2 Uninformed Search Strategies **(7 hours)**

Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies. [CO3] [PO3, PSO1]

Unit 3 Informed Search Methods: **(7 hours)**

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Tabu Search, Constraint satisfaction, Means Ends Analysis, **Game playing:** Minimax Search, Alpha-Beta Cut offs, Waiting for Quiescence. [CO3, CO6] [PO3]

SECTION-II

Unit 4 Local Search Algorithms and Optimization Models **(7 hours)**

Iterated Hill Climbing, Simulated Annealing, Local Beam Search, Evolutionary Algorithms - Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization. [CO4] [PO2]

Unit 5 Planning: **(7 hours)**

Blocks world, STRIPS, Implementation using goal stack, **Planning with state space search:** Forward state space search, Backward state space search, Heuristics for state space search. Partial Order Planning, Planning Graphs, Hierarchical planning, Least commitment strategy. Conditional Planning, Continuous Planning. [CO5] [PO4]

Unit 6 Natural Language Processing and Anatomy of building AI systems **(7 hours)**

Process, Syntax & Semantics, Disambiguation & Information retrieval
Shortcomings of AI, Building Fair models, Interpretable models Hadoop & Big Data

List of Practical's: (Any ….)

1. Implementation of AI and Non-AI technique by implementing any two player game [CO1,CO2] [PO1,PO2]
2. Implementation of Uninformed Search strategies [CO1, CO2] [PO1, PO2]
3. Implementation of Informed Search strategies [CO2, CO3] [PO2, PO3]
4. Implementation of CSP Problem [CO3] [PO3]
5. Implement Local Search Techniques using GA/ACO/PSO [CO5] [PO4]

Text Books: (As per IEEE format)

1. Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill

2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.

3. Deepak Khemani: "A First Course in Artificial Intelligence", Mc Graw Hill

4. Saroj Kaushik: "Artificial Intelligence" Cengage Publication

Reference Books: (As per IEEE format)

1. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley

2. Patterson: "Introduction to AI and Expert Systems", PHI

3. Nilsson: "Principles of Artificial Intelligence", Morgan Kaufmann.

Course Outcomes:

The student will be able to –

1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents capable of problem formulation.
2. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
3. Evaluation of different uninformed and informed search algorithms on well formulated problems along with stating valid conclusions that the evaluation supports.
4. Design and implement the local search algorithms like Hill Climbing, simulated annealing and evolutionary techniques like Genetic algorithm, Ant Colony Optimization, particle swarm optimization for designing solution for the problems like N-Queen problem or Travelling Salesman problem.
5. Analyze the AI problem using different planning techniques.
6. Understand the importance of ethical considerations while designing AI solutions.

Moocs Links and additional reading material: (If any)

www.nptelvideos.in

Future Courses Mapping:

1. Advanced Machine Learning
2. Deep Learning
3. Distributed Artificial Intelligence
4. Edge AI and Edge Computing
5. Privacy-Preserving Machine Learning
6. Trustworthy and Explainable AI

Job Mapping: (Optional)

- **AI/ML Engineer**
- **Data Scientist**
- **Machine Learning Engineer**
- **Deep Learning Engineer**
- **Federated Learning Engineer**
- **Edge AI Engineer**
- **AI Research Engineer**
- **Computer Vision Engineer**
- **Research Associate (Artificial Intelligence)**

CO - PO Mapping:

CO	Program Outcomes (PO)												PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CS3202.1	2															
CS3202.2		2														
CS3202.3			2										1			
CS3202.4		1														
CS3202.5				3												
CS3202.6			1													
Average	2	3	3	3									1			

FF No.: 654

AI 3202: Machine Learning

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	Lab Assessment: 10 Marks
Tutorial :	-- Hrs/week	ESE(W) : 40 Marks

Course Prerequisites:

- Linear Algebra, Statistics, Probability, Calculus, and Programming Languages

Course Objectives:

The course aims to:

1. Introduce the fundamental concepts of machine learning, learning paradigms, concept learning, and regression techniques.
2. Provide knowledge of supervised learning algorithms and model evaluation techniques for classification and prediction.
3. Familiarize students with Bayesian, ensemble, and kernel-based learning techniques for machine learning applications.
4. Expose students to unsupervised learning, association rule mining, dimensionality reduction, reinforcement learning, and artificial neural networks.
5. Develop the ability to apply machine learning techniques using modern software tools for solving real-world data-driven problems.

Course Relevance:

Machine Learning is a fundamental discipline of Artificial Intelligence (AI) that enables computer systems to learn patterns from data and make predictions or decisions without being explicitly programmed for every task. It provides the foundation for developing intelligent systems capable of solving complex real-world problems through data-driven learning and adaptation.

The rapid advancement of Machine Learning has transformed diverse domains such as healthcare, finance, manufacturing, retail, agriculture, transportation, cybersecurity, and smart cities by enabling predictive analytics, automation, recommendation systems, fraud detection, and intelligent decision support. The techniques covered in this course equip students with the knowledge and practical skills required to design, develop, and evaluate machine learning models using modern tools and methodologies.

This course provides a strong foundation for advanced studies in Artificial Intelligence, Deep Learning, Natural Language Processing, Computer Vision, and Data Science, while preparing students for careers in intelligent systems development, data analytics, and AI-driven application domains.

SECTION-I

UNIT 1: **(4 Hours)**

Types of Learning: Supervised, Unsupervised, Reinforcement. Concept Learning, General-to-Specific Ordering, Find S algorithm, Version space and the candidate elimination algorithm, inductive bias, Bias, Variance, Underfitting, Overfitting. Linear Regression, Logistic Regression, Inductive learning.

UNIT 2: **(4 Hours)**

Decision Tree Learning: Representation, Basic decision tree learning algorithm, Issues in decision tree learning, and Random Forest Model. Model Evaluation: Train -Test Split, Cross-Validation, Confusion Matrix, Accuracy, Precision, Recall and F1-score.

UNIT 3: **(6 Hours)**

Bayesian Learning: Probability, Bayesian Learning: Bayes theorem, Naïve Bayes algorithm, Maximum Likelihood Estimation. Ensemble Learning: Bagging and boosting. SVM: Kernel functions, Linear SVM, Nonlinear SVM, Hyperparameter tuning, Handling Imbalanced Datasets, k-Nearest Neighbors (k-NN)

SECTION-II

UNIT 4: **(5 Hours)**

Clustering Algorithms: Unsupervised learning, clustering. Partition based clustering, K-means and K-Medoids, Hierarchical clustering, Density based clustering algorithms.

UNIT 5: **(5 Hours)**

Association rules mining: Apriori Algorithm, Support and Confidence Measures, Introduction to Hidden Markov model, Dimensionality Reduction Techniques: Principal Component Analysis (PCA) and Singular Value Decomposition (SVD).

UNIT 6: **(4 Hours)**

Reinforcement learning: Exploration, Exploitation, Rewards, Penalties, Markov Decision Process, Q-Learning and Bellman Equation. Artificial Neural Networks: Basics of ANN, Feedforward Neural Networks,

Introduction to Deep Neural Networks.

List of Assignments:

1. Survey on Explainable AI (XAI): Techniques, Challenges, and Trends
2. A Comparative Study of Large Language Models (LLMs) and Their Fine-tuning Techniques
3. Evolution of Transfer Learning in Computer Vision: From AlexNet to Vision Transformers
4. Survey on Federated Learning: Privacy-Preserving Machine Learning at the Edge
5. Review of Recent Advances in Self-Supervised Learning
6. Case Study on ML in Credit Scoring: Risk vs Fairness
7. Propose a Personalized News Recommendation System Using Reinforcement Learning
8. Design a Smart Healthcare Assistant Using Multi-Modal Learning (Text + Imaging)
9. Architecture of an ML-Enabled Chatbot for Student Support Systems in Universities
10. From Model Accuracy to Model Fairness: What Should We Prioritize?
11. The Ethics of Synthetic Data: Solution or New Problem

List of Tutorials:

1. Feature Selection Techniques
2. Supervised Learning
3. Unsupervised Learning
4. Reinforcement Learning
5. SVM
6. Item based Recommender system
7. Shallow Neural Networks
8. Key concepts on Deep Neural Networks
9. Practical aspects of deep learning, Optimization Algorithms
10. Hyperparameter tuning, Batch Normalization, Programming Frameworks
11. Bird recognition in the city of Peacetopia (case study)
12. Autonomous driving (case study)
13. The basics of ConvNets
14. Detection Algorithms
15. Special Applications: Face Recognition & Neural Style Transfer
16. Natural Language Processing and Word Embeddings
17. Sequence Models and Attention Mechanism

List of Practical's:

1. Develop the Find-S Algorithm for concept learning using a suitable dataset.
2. Build a Simple Linear Regression model to predict house prices using a housing dataset.
3. Develop a Multiple Linear Regression model to predict product sales using advertising expenditure data.
4. Design a Decision Tree Classification model for loan approval prediction and evaluate its performance using appropriate evaluation metrics.
5. Develop a Random Forest Classification model for insurance fraud detection and compare its performance with the Decision Tree classifier.
6. Build a Naïve Bayes classifier for spam email detection.
7. Develop a Support Vector Machine (SVM) classifier for diabetes prediction using a healthcare dataset.
8. Build a k-Nearest Neighbors (k-NN) classifier for diabetes prediction and analyze the effect of different values of *k* on classification accuracy.
9. Apply K-Means Clustering for customer segmentation using retail data.
10. Perform Hierarchical Clustering and compare the clustering results with K-Means.
11. Apply Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) for dimensionality reduction and data visualization.
12. Perform Association Rule Mining using the Apriori Algorithm to discover frequent itemsets and generate association rules.
13. Develop an Artificial Neural Network (ANN) for customer churn prediction or credit card fraud detection.
14. Develop and evaluate a machine learning solution for a real-world application using an appropriate algorithm. Compare model performance using suitable evaluation metrics and analyze Ethical AI aspects such as fairness, bias, privacy, explainability, and societal impact.

List of Course Projects:

Following types of problem statements can be taken for course project.

1. Sentiment Analysis of Movie or Restaurant Reviews
2. Heart Disease Prediction using Machine Learning
3. Market Basket Analysis
4. Credit Card Fraud Detection
5. Handwritten Digit Recognition
6. Image Caption Generation
7. Movie Recommendation System
8. Cancer Classification
9. Traffic Sign Recognition
10. Customer Segmentation using Machine Learning
11. Uber Data Analysis
12. Loan Prediction
13. HVAC Demand Forecasting

14. Customer Relationship Management using Machine Learning
15. Clinical Decision Support System
16. Insurance Claim Fraud Detection
17. Portfolio and Stock Price Prediction
18. Smart Building Energy Management System
19. Quality Analysis of Cereals, Oilseeds, and Pulses using Machine Learning
20. Building a Recurrent Neural Network (RNN) for Sequence Prediction
21. Operations on Word Vectors for Natural Language Processing
22. Neural Machine Translation with Attention Mechanism
23. Traffic Signal Control using Reinforcement Learning
24. Autonomous Robot Path Planning using Reinforcement Learning
25. Demand Forecasting using Gradient Boosting or XGBoost
26. Predictive Maintenance of Industrial Equipment
27. Crop Yield Prediction using Environmental Data
28. Air Quality Prediction using Machine Learning
29. Disease Outbreak Prediction using Machine Learning
30. Explainable AI (XAI) for Model Interpretation and Fairness Analysis

List of Course Seminar Topics:

1. Validation
2. Naive Bayes Algorithm
3. Machine and Privacy
4. Limitations of ML
5. Ensemble Learning
6. Dimensionality reduction algorithms
7. Comparison of Machine Learning algorithms
8. Feature Extraction In Machine Learning
9. Reinforcement Learning
10. Probabilistic Model
11. Dropout: a simple way to prevent neural networks from overfitting,
12. Deep Residual Learning for Image Recognition
13. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift
14. Large-Scale Video Classification with Convolutional Neural Networks
15. Generative adversarial nets
16. High-Speed Tracking with Kernelized Correlation Filters
17. Do we need hundreds of classifiers to solve real world classification problems
18. A survey on concept drift adaptation

Text Books: (As per IEEE format)

1. T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.

2. P. Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge, U.K.: Cambridge University Press, 2012.

Reference Books: (As per IEEE format)

1. E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020.
2. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2012.
3. C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

Course Outcomes:

After completion of the course, student will be able to –

1. Explain machine learning paradigms and apply concept learning and regression techniques.
2. Apply decision tree-based classification algorithms and evaluate model performance using appropriate evaluation metrics.
3. Apply Bayesian learning, ensemble learning, Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) for classification and prediction.
4. Compare partition-based, hierarchical, and density-based clustering methods for unsupervised learning.
5. Apply association rule mining and dimensionality reduction techniques, and explain the fundamentals of Hidden Markov Models.
6. Explain the fundamental concepts of reinforcement learning and artificial neural networks.

Moocs Links and additional reading material:

- https://www.coursera.org/learn/machine-learning
- https://cs229.stanford.edu/
- https://ocw.mit.edu/courses/6-036-introduction-to-machine-learning-fall-2020/
- https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/
- https://thegradient.pub/
- https://distill.pub/

Future Courses Mapping:

1. Deep Learning
2. Natural Language Processing
3. Computer Vision
4. Reinforcement Learning
5. Federated Learning
6. Explainable and Responsible AI
7. Large Language Models (LLMs) and Prompt Engineering

8. AI Agents and Autonomous Systems

Job Mapping:

Job opportunities that one can get after learning this course

1. Machine Learning Engineer
2. Data Scientist
3. Data Analyst
4. AI Engineer
5. Business Intelligence (BI) Analyst

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	3	3	2									1	2	
CO2	3	2	3	3	3	2		1	2	1		1		2
CO3	3	3	3	2	3	2			2	1			2	
CO4	3	2		2	2				2	1			2	
CO5	3	2	2	2	3	2		1				1		2
CO6	2	2	2	2	2				2	1		1		2
Average	**2.8**	**2.3**	**2.4**	**2.2**	**2.6**	**2**		**1**	**2**	**1**		**1**	**2**	**2**

FF No.: 654

AI 3206A: Operating System

Teaching Scheme		Examination Scheme
Credits:	4	CP: 40 Marks
Lectures :	3 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	GD/PPT: 20 Marks
Tutorial :	-- Hrs/week	HA: 20 Marks

Course Prerequisites:

1. Basics of Computer System
2. Computer Organization
3. Data Structures
4. Any Programming Language.

Course Objectives:

1. To understand the basic concepts and functions of Operating Systems.
2. To gain knowledge of process synchronization and its mechanism.
3. To get familiar with CPU scheduling algorithms.
4. To discuss different deadlock handling mechanisms.
5. To learn memory management techniques and virtual memory.
6. To evaluate various disk scheduling algorithms

Course Relevance:

This course focuses on functions of operating systems. Operating system is a System software that manages the resources of the computer system and simplifies applications programming. The Operating System acts as a platform of information exchange between your computer's hardware and the applications running on it

SECTION-I

Unit 1: (Introduction to Operating System) **6 Hours**

What is Operating System? Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of System calls, Types of OS: Batch, Multiprogramming, Time Sharing, Parallel, Distributed & Real-time OS. Developments leading to Modern Operating Systems. Introduction to Linux OS, BASH Shell Scripting: Basic shell commands.

Unit 2: (Process Management) **8 Hours**

Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control, Multithreading models, Thread implementations – user level and kernel level threads, Concurrency: Issues with concurrency, Principles of Concurrency, Mutual Exclusion: OS/Programming Language Support: Semaphores, Mutex, Classical Process

Synchronization problems.

Unit 3: (Uniprocessor Scheduling) **7 Hours**

Types of Scheduling algorithms FCFS, SJF, RR, Priority

SECTION-II

Unit 4: (Deadlock) **6 Hours**

Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery Example: Dining Philosophers Problem / Banker's Algorithm.

Unit 5: (Memory Management) **8 Hours**

Memory Management requirements, Memory Partitioning, Paging, Segmentation, Address translation, Placement Strategies: First Fit, Best Fit, Next Fit and Worst Fit. Virtual Memory, VM with Paging, VM with Segmentation, Page Replacement Policies: FIFO, LRU, Optimal

Unit 6: (I/O & File Management) **7 Hours**

I/O Devices - Types, Characteristics of devices, I/O Buffering. Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN

Overview-Files and File Systems, File structure. File Organization and

Access, File Directories, File Sharing, Record Blocking, Secondary Storage Management

List of Practical's: (Any ….)

1. Execution of Basic Linux commands.
2. Execution of Advanced Linux commands.
3. Any shell scripting program.
4. Write a program demonstrating use of different system calls.
   e. g FORK, EXECVE and WAIT system calls along with zombie and orphan states.
5. Implement multithreading for Matrix Operations using Pthreads. e.g User level and kernel level threads.
6. Implementation of Classical problems using Threads and Mutex.
7. Implementation of Classical problems using Threads and Semaphore.
8. Write a program to compute the finish time, turnaround time and waiting time for the following algorithms:
   a. First come First serve
   b. Shortest Job First (Preemptive and Non-Preemptive)
   c. Priority (Preemptive and Non-Preemptive)
   d. Round robin
9. Write a program to check whether given system is in safe state or not using Banker's Deadlock Avoidance algorithm.

10. Write a program to calculate the number of page faults for a reference string for the following page replacement algorithms:

a. FIFO b) LRU c) Optimal

AI Based Assignments

1. AI-Driven CPU Scheduling Simulator

- **Objective**: Implement classical CPU scheduling algorithms (FCFS, SJF, RR, etc.), then build a machine learning model to predict the best scheduling algorithm to use based on workload characteristics.
- **Skills**: OS scheduling, data collection, classification models (e.g., decision trees).
- **Bonus**: Use reinforcement learning to learn optimal scheduling policies.

---

2. Memory Management Prediction Using ML

- **Objective**: Simulate memory management with paging. Collect access patterns and train a model to predict page faults or optimal page replacement.
- **AI Component**: Use LSTM or other sequence models to predict next page accesses.
- **OS Component**: Implement LRU, FIFO, and compare with ML prediction.

---

3. Anomaly Detection in System Logs

- **Objective**: Parse system logs (e.g., dmesg, syslog) and detect unusual behavior using unsupervised learning.
- **AI Component**: Use K-Means or Autoencoders for anomaly detection.
- **Skills**: Log parsing, unsupervised learning, basic NLP.

---

4. AI-Based Deadlock Prediction

- **Objective**: Simulate a resource allocation graph and train a classifier to predict potential deadlocks based on system state.
- **ML Model**: Decision trees or SVMs trained on past resource allocation states.
- **OS Concept**: Deadlock detection/prevention.

List of Tutorials:

List of Group Discussion Topics:

1. Flynn’s taxonomy
2. Role of Operating system

3. 32 bit Vs 64 bit OS
4. Storage structures and their tradeoffs
5. Disk Scheduling
6. Desktop OS Vs Mobile OS
7. Security Vs Protection in OS
8. I/O processors
9. Linux Vs Windows OS
10. Best OS for smartphones

List of Home Assignments:

Design:

1. Report Generation using Shell Script and AWK
2. Library Management System using shell
3. Inter Process Communication in Linux
4. Design any real time application using job scheduling
5. Design any application using Android

Case Study:

1. Distributed Operating System
2. Microsoft Windows 11
3. VMware
4. Linux
5. Android

Surveys:

1. A survey of Desktop OS
2. Analysis and Comparison of CPU Scheduling Algorithms
3. Device Drivers for various devices
4. Parallel Computing
5. Malware Analysis, Tools and Techniques

Blog

1. Operating System Forensics
2. Open Source OS Vs Commercial OS
3. BIOS
4. Comparative study of different mobile OS
5. Operating Systems for IoT Devices

List of Course Projects:

1. Design and implementation of a Multiprogramming Operating System: Stage I
   i. CPU/ Machine Simulation
   i. Supervisor Call through interrupt
2. Design and implementation of a Multiprogramming Operating System: Stage II
   i. Paging
   ii. Error Handling
   iii. Interrupt Generation and Servicing
   iv. Process Data Structure
3. Design and implementation of a Multiprogramming Operating System: Stage III
   i. Multiprogramming
   ii. Virtual Memory
   iii. Process Scheduling and Synchronization
   iv. Inter-Process Communication
   v. I/O Handling, Spooling and Buffering

List of Course Seminar Topics: (*if applicable to your course*)

1. Different File Systems in Windows and Linux OS
2. Operating System generations
3. OS Structures
4. HDFS
5. Process Vs Threads
6. Virtual Machines
7. Real Time Scheduling
8. Booting Process of different Operating Systems.
9. RAID
10. Protection and Security in Operating System

Text Books: (As per IEEE format)

1. *Stalling William; "Operating Systems"; 6th Edition, Pearson Education*
2. *Silberschatz A., Galvin P., Gagne G.;"Operating System Concepts" ; 9th Edition; John Wiley and Sons;*
3. *Yashavant Kanetkar; "Unix Shell Programming"; 2nd Edition, BPB Publications*
4. *Sumitabha Das; "Unix Concepts and Applications"; 4th Edition, TMH.*
5. *D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN – 0074635794*
6. *John J Donovan; "Systems Programming"; Tata Mc-Graw Hill Edition, ISBN-13978- 0-07-*

460482-3

Reference Books: (As per IEEE format)

1. *Silberschatz A., Galvin P., Gagne G; "Operating System Principles"; 7th Edition, John Wiley and Sons.*
2. *Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming"; 1st Edition, Australia Thomson Brooks Cole.*
3. *Achyut S. Godbole , Atul Kahate; "Operating Systems"; 3rd Edition, McGraw Hill.*

Course Outcomes:

The student will be able to –

Upon completion of the course, student will be able to –

1. Examine the functions of a contemporary Operating System with respect to convenience, efficiency and the ability to evolve.
2. Demonstrate knowledge in applying system software and tools available in modern operating systems for process synchronization mechanisms.
3. Apply various CPU scheduling algorithms to construct solutions to real world problems.
4. Identify the mechanisms to deal with Deadlock.
5. Illustrate the organization of memory and memory management techniques
6. Acquire a detailed understanding of various I/O buffering techniques and disk scheduling algorithms

Moocs Links and additional reading material: (If any)

1. www.nptelvideos.in
2. https://www.udemy.com/
3. https://learn.saylor.org/
4. https://www.coursera.org/
   https://swayam.gov.in/

Future Courses Mapping:

1. Advance Operating System
2. Unix Operating System
3. Linux programming

4. 4. Distributed System/Computing System Programming

Job Mapping: (Optional)

Job opportunities that one can get after learning this course

1. Linux Administration

2. Kernel Developers

3. Application Developers

4. System programmer

5. System architect

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3									1				
CO2		3											2	
CO3			3										2	
CO4		3												
CO5	3												2	
CO6					3								2	
Average	1	1	0.5		0.5					0.16			1.3	

FF No.: 654

AI3206B: System Programming

Teaching Scheme	Examination Scheme
Credits: 04	CP: 40 Marks
Lectures : 03 Hrs/week	CVV: 20 Marks
Practical : 02 Hrs/week	GD/PPT:20 Marks
Tutorial : 00 Hrs/week	HA :20 Marks

Course Prerequisites:

- Basic knowledge of C/C++ programming.
- Fundamentals of Computer Organization and Architecture.
- Understanding of Data Structures.
- Familiarity with Discrete Mathematics, finite automata, and formal grammars.
- Basic problem-solving and algorithm development skills.

Course Objectives:

- To understand the fundamentals of system programming and system software components.
- To learn the design and implementation of assemblers, macro processors, loaders, and linkers.
- To understand the concepts of programming language design and language translation.
- To study compiler phases, lexical analysis, syntax analysis, and parser design.
- To develop analytical and problem-solving skills in system software development.
- To apply system programming concepts for designing efficient language processing systems.

Course Relevance:

- Provides a strong foundation in system software and its interaction with computer hardware.
- Develops an understanding of assemblers, macro processors, loaders, linkers, and compilers.
- Enhances knowledge of program translation, memory management, and executable generation.
- Introduces compiler design concepts, including lexical analysis, syntax analysis, finite automata, and parsing techniques.
- Strengthens analytical, debugging, and problem-solving skills required for system-level software development.

SECTION-I

Unit 1: Introduction (7 Hours)

Introduction to Systems Programming, Need of systems programming, Software Hierarchy, Types of software: system software and application software, Machine structure. Evolution of components of

systems programming: Text Editors, Assembler, Macros, Compiler, Interpreter, Loader, Linker, Debugger, Device Drivers.

Unit 2: Assembler **(7 Hours)**

Elements of Assembly Language Programming: Assembly Language statements, Benefits of Assembly Language, A simple Assembly scheme, Pass Structure of Assembler. Design of two pass assembler: Processing of declaration statements, Assembler Directives and imperative statements, Advanced Assembler Directives, Intermediate code forms, Pass I and Pass II of two pass Assembler.

Unit 3: Macro Language and Macro Processor **(7 Hours)**

Introduction, Features of a Macro facility: Macro instruction arguments, Conditional Macro expansion, Macro calls within Macros, Macro instructions, Defining Macro, Design of two pass Macro processor, Concept of single pass Macro processor

SECTION-II

Unit 4: Linkers and Loaders **(7 Hours)**

Introduction, Loader schemes: Compile and Go, General Loader Scheme, Absolute Loaders, Subroutine Linkages, Relocating Loaders, Direct linking Loaders, Overlay structure, Design of an Absolute Loader, Design of Direct linking Loader, Self-relocating programs, Static and Dynamic linking.

Unit 5: Programming Languages **(7 Hours)**

Importance of High Level Languages, Features of High Level Language, Data Types and Data Structures, Storage Allocation and Scope Names, Accessing Flexibility, Functional Modularity, Asynchronous Operation, Extensibility and Compile-Time Macros.

Unit 6: Compilers **(7 Hours)**

Language Translation- introduction, basics, steps involved in atypical language processing system, Types of translators, Compilers- overview, phases, Pass and Phases of translation, bootstrapping, data structures in compilation. Lexical Analysis (Scanning)- Functions of scanner, Specification of tokens Regular expressions and Regular grammars for common PL constructs. Recognition of Tokens- Finite Automata in recognition and generation of tokens. Scanner generators- Lexical analyzer generators, LEX. Syntax Analysis (Parsing)- Functions of a parser, Classification of parsers. Context free grammars in syntax specification, benefits and usage in compilers.

List of Practical's: (Any ….)

1. Design and implement Pass-I of a Two-Pass Assembler to generate Symbol Table (SYMTAB), Literal Table (LITTAB), Pool Table (POOLTAB), and Intermediate Code (IC).

2. Design and implement Pass-II of a Two-Pass Assembler that reads the intermediate code and symbol table generated by Pass-I and produces the final machine code.

3. Design and implement Pass-I of a Two-Pass Macro Processor to generate Macro Name Table (MNT), Macro Definition Table (MDT), Argument List Array (ALA), and Keyword Parameter Default Table (KPDTAB).

4. Design and implement Pass-II of a Two-Pass Macro Processor to expand macro calls using the tables generated in Pass-I and produce the expanded assembly program.

5. Design and implement an Absolute Loader that loads an object program into memory and displays the memory map after loading.

6. Design and implement a Direct Linking Loader that performs relocation and linking of multiple object modules and generates the final executable memory image.

7. Design and implement a Lexical Analyzer that identifies keywords, identifiers, constants, operators, delimiters, and comments from a source program.

8. Design and implement a LEX program to generate a lexical analyzer for recognizing tokens of a programming language.

9. Write a program using YACC specifications to implement syntax analysis phase of compiler

to validate type and syntax of variable declaration in Java.

10. Write a program using YACC specifications to implement syntax analysis phase of compiler to recognize simple and compound sentences given in input file.

11. Write a LEX and YACC program to recognize and evaluate arithmetic expressions with appropriate operator precedence and syntax error handling.

List of Tutorials:

List of Course Projects:

- Mini Compiler for a Simple Programming Language
- Two-Pass Assembler
- Two-Pass Macro Processor
- Lexical Analyzer and Parser using LEX/YACC
- Static Source Code Analyzer with Tokenization and Error Detection

List of Course Seminar Topics: (*if applicable to your course*)

- Compiler Design: Phases and Applications
- LLVM Compiler Infrastructure
- Static Linking vs Dynamic Linking
- Design of Two-Pass Assembler
- Macro Processor Architecture
- Finite Automata in Lexical Analysis
- LEX and YACC for Compiler Construction
- Web Assembly: The Next Generation of Portable Code
- AI-Based Compiler Optimization Techniques
- Role of System Programming in Operating Systems

Text Books: (As per IEEE format)

1. D. M. Dhamdhere, Systems Programming and Operating Systems, McGraw-Hill Education, 2nd Edition.
2. Leland L. Beck, System Software: An Introduction to Systems Programming, Pearson Education, 3rd Edition.

Reference Books: (As per IEEE format)

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, Pearson, 2nd Edition.
2. Keith D. Cooper and Linda Torczon, Engineering a Compiler, Morgan Kaufmann, 2nd Edition.
3. Andrew W. Appel, Modern Compiler Implementation in C, Cambridge University Press.
4. Steven S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufmann.
5. Michael L. Scott, Programming Language Pragmatics, Morgan Kaufmann, 4th Edition.
6. M. Morris Mano, Computer System Architecture, Pearson, 3rd Edition.

Course Outcomes:

The student will be able to –

CO1. Explain the fundamentals of system programming, software hierarchy, machine structure, and the evolution of system software components.
CO2. Analyze the design and implementation of a two-pass assembler, including assembler directives, symbol table generation, intermediate code generation, and Pass-I & Pass-II processing.
CO3. Apply the concepts of macro processors to design and implement single-pass and two-pass macro processors with macro expansion techniques.
CO4. Analyze various loader and linker schemes, relocation techniques, overlay structures, and static and dynamic linking mechanisms.
CO5. Explain the features of high-level programming languages, storage allocation techniques, data structures, scope management, and compile-time macro concepts.
CO 6. Analyze the phases of compiler design, including lexical analysis, syntax analysis, finite automata, regular expressions, context-free grammars, and parser generation techniques.

Moocs Links and additional reading material: (If any)

- https://www.coursera.org/in/articles/system-programming
- https://www.coursera.org/learn/linux-system-programming-introduction-to-buildroot
- https://www.youtube.com/watch?v=teO316cTQ0Y
- https://www.youtube.com/watch?v=KEQZodb1m_A

Future Courses Mapping:

- Compiler Design
- Distributed Systems
- Embedded Systems

Job Mapping: (Optional)

- System Programmer
- Compiler Engineer
- Software Developer
- AI Software Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	-	-	-	-	-	-	-	-	-	1	2	-
CO2	3	3	2	-	2	-	-	-	-	-	-	1	3	1
CO3	3	2	3	-	3	-	-	-	1	-	-	1	3	2
CO4	3	3	2	1	2	-	-	-	-	-	-	1	2	2
CO5	3	2	1	-	2	-	-	-	-	-	-	1	2	1
CO6	3	3	2	2	3	-	-	-	-	1	-	2	3	2
Average	3	2.5	2	1.5	2.4	-	-	-	1	1	-	1	2.5	1.6

FF No.: 654

AI3206C: MLOPs

Teaching Scheme	**Examination Scheme**
Credits: 4	CP: 40 Marks
Lectures : 3 Hrs/ Week	GD/PPT: 20 Marks
Practical : 2 Hrs/ Week	HA: 20 Marks
Tutorial : NA	CVV: 20 Marks

Course Prerequisites:

Artificial Intelligence, Machine Learning, Deep Learning

Course Objectives:

1.Understand the basics of MLOps and ML lifecycle.

2.Learn data versioning, experiment tracking, and automation concepts.

3. Build simple ML pipelines using industry tools.
4. Deploy machine learning models using APIs and containers.
5. Understand monitoring, Explainability, and responsible AI concepts.
6. Develop practical skills for production-ready ML systems.

Course Relevance:

Machine Learning Operations (MLOps) is an essential discipline that bridges the gap between machine learning model development and real-world deployment. It combines Machine Learning, DevOps, and Cloud Computing practices to automate, deploy, monitor, and maintain AI systems efficiently. As industries increasingly adopt AI-driven solutions, professionals with MLOps skills are in high demand. This course equips students with the knowledge and practical skills required to build scalable, reliable, and production-ready machine learning systems for modern industry applications.

SECTION-I

Unit 1:Introduction to MLOps & ML Lifecycle (7 Hours)

Difference between Software Dev Lifecycle vs. ML Lifecycle

Introduction to MLOps: Definition, Need, Challenges.

Key Concepts: Versioning, Reproducibility, Automation, Monitoring

Stages of ML Lifecycle: Problem Formulation, Data Collection and Preprocessing, Model Development, Training, Evaluation, Model Deployment and Monitoring, Overview of MLOps Architecture and Workflow,

Overview of popular MLOps platforms and cloud services.

Case Studies: Student Performance Prediction System using complete ML workflow

Unit-II Data, Feature, and Experiment Management 07 Hours

Introduction to Data Versioning Concepts, DVC Basics

Introduction to Dataset Management

Experiment Tracking using MLflow,

Feature Engineering Concepts: Encoding, Scaling, Normalization

Model Registry Concepts (e.g., MLflow Model Registry)

Need for Reproducibility in ML

Basics of Data Validation

Case Studies: Track ML experiments using version control and experiment tracking tools.

Unit-III ML Pipeline Automation 07 Hours

Introduction to ML Pipelines, Workflow Automation Concepts

Concept of Pipelines & DAGs, Tools Overview: Apache Airflow,

Automating: Data Pre-processing, Feature Engineering, Model Training & Evaluation

Basics of CI/CD for Machine Learning, Basic testing and validation concepts in ML pipelines.

Case Studies: Build simple workflow automation using Airflow

SECTION-II

Unit-IV Model Deployment and Serving **07 Hours**

Introduction to Model Deployment

REST API using FastAPI, Docker basics for ML deployment

Batch vs Real-time inference

Introduction to monitoring, Logging and alerting basics

Basic concepts of drift detection

Case Studies: Containerize ML model using Docker

Unit-V Responsible and Explainable AI Systems **07 Hours**

Introduction to Responsible AI, Bias and Fairness in ML

Explainability using SHAP and LIME, Basic data privacy concepts

Ethical issues in AI systems, Governance basics

Case Studies: Monitoring ML models and understanding model drift concepts.

Unit-VI Cloud-based MLOps and Industry Applications **07 Hours**

Introduction to Cloud-based MLOps , Overview of cloud ML services,

Introduction to AWS SageMaker: Studio, Training Jobs, Model Deployment, and Monitoring

MLOps use cases in healthcare, agriculture, finance

Best practices in ML deployment, End-to-end MLOps workflow case study

Case Studies: MLOps in Agriculture / Healthcare / Finance, Best Practices in Industrial MLOps Workflows

List of Assignments:

1. Implement dataset versioning using DVC
2. Track ML experiments using MLflow
3. Create feature engineering pipeline
4. Build workflow automation using Airflow
5. Create REST API using FastAPI
6. Containerize ML model using Docker
7. Implement CI/CD pipeline using GitHub Actions
8. Monitor model using Prometheus and Grafana
9. Apply SHAP/LIME for Explainability
10. Train and deploy a Machine Learning model using AWS SageMaker
11. Mini project on complete MLOps workflow

List of Course Projects:

1. Setting up a complete MLOps pipeline

2. Deploying a machine learning model to production

3. Monitoring and maintaining a deployed model

4. Implementing CI/CD for a machine learning project

List of Home Tutorial:

1. Introduction to MLOps and Machine Learning Lifecycle

   Study the concepts, need, challenges, and stages of MLOps.

2. Experiment Tracking and Reproducibility Using MLflow

   Learn to record parameters, metrics, artifacts, and experiment history.

3. Dataset Versioning Using DVC

   Study dataset version control and reproducible machine learning workflows.

4. Feature Engineering and Data Processing Pipelines

   Implement feature transformation techniques such as scaling, normalization, and encoding.

5. Workflow Orchestration Using Airflow/Kubeflow

   Study DAG concepts, task scheduling, and workflow automation.

6. Model Packaging and Containerization Using Docker

   Learn model serialization, dependency management, and container creation.

7. CI/CD Pipeline Development for Machine Learning Using GitHub Actions

   Explore continuous integration, delivery, and training workflows.

8. Model Monitoring and Drift Detection Using Prometheus and Grafana

   Implement monitoring dashboards and drift detection techniques.

List of Course Seminar Topics:

1. Cloud-Native MLOps Architectures for Large-Scale Machine Learning Systems

2. Feature Stores for Real-Time and Batch Machine Learning Pipelines
3. Hyperparameter Optimization and AutoML Techniques in MLOps
4. Edge AI Deployment and Challenges in Resource-Constrained Environments
5. Distributed Machine Learning and High-Performance Computing (HPC) Systems
6. Large Language Model Operations (LLMOps): Concepts and Applications
7. AI Governance, Model Risk Management, and Regulatory Compliance
8. Advanced Model Monitoring: Drift Detection, Observability, and Automated Retraining
9. Serverless Machine Learning Deployment for Scalable Applications
10. AI and Explainable Machine Learning in Production Systems

Text Books: (As per IEEE format)

1. M. Treveil and A. Shukla, Introducing MLOps: How to Scale Machine Learning Projects with Continuous Delivery, Monitoring, and Governance. Sebastopol, CA, USA: O'Reilly Media, 2020.

2. D. Brinkmann, E. Raj, and M. Daoust, MLOps Engineering at Scale. Sebastopol, CA, USA: O'Reilly Media, 2023.

Reference Books: (As per IEEE format)

1. C. Osipov, MLOps: Continuous Delivery and Automation Pipelines in Machine Learning. Shelter Island, NY, USA: Manning Publications, 2022.

2. N. Gift and A. Deza, Practical MLOps. Sebastopol, CA, USA: O'Reilly Media, 2021.

3. E. Ameisen, Building Machine Learning Powered Applications. Sebastopol, CA, USA: O'Reilly Media, 2020.

Course Outcomes:

The student will be able to –

1. To explain the key differences between software development and ML lifecycles.
2. To apply MLOps tools like MLflow, DVC in workflows.
3. To manage data, features, and experiments effectively.
4. To automate ML pipelines using CI/CD integration.
5. To deploy and monitor ML models with appropriate tools.
6. To implement responsible AI practices, ensuring fairness, privacy, and compliance.

Moocs Links and additional reading material:

1. Practical MLOps (Coursera - DeepLearning.AI)

   Link: https://www.coursera.org/specializations/mlops-machine-learning-duke

2. MLOps Specialization (DeepLearning.AI - Coursera)

   Link: https://www.coursera.org/courses

3. NPTEL - DevOps and MLOps (IIT Ropar)
4. End-to-End Machine Learning with MLOps (Google Cloud)

   Link: https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

5. MLflow: End-to-End Machine Learning Lifecycle (Databricks)

   Link: https://docs.databricks.com/gcp/en/mlflow/end-to-end-example

6. Kubernetes for Machine Learning (edX)

   Link: https://www.edx.org/learn/kubernetes

Future Courses Mapping:

1. Advanced Artificial Intelligence
2. Edge AI
3. Cloud Computing and Distributed Systems

Job Mapping:

1. AI/ML Engineer
2. IoT Engineer
3. Embedded AI Engineer
4. MLOps Engineer
5. Cloud & Devop Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	--	2	--	--	--	--	1	--	2	1	1
CO2	2	3	2	1	3	--	--	--	--	1	--	2	2	2
CO3	2	2	3	2	3	--	--	--	1	1	1	2	2	3
CO4	2	2	3	2	3	--	--	--	1	2	1	2	2	3
CO5	2	3	2	2	3	--	--	1	1	2	1	2	3	3
CO6	2	3	1	2	2	2	1	3	1	1	--	3	2	2
Average	2.2	2.5	2.0	1.5	2.7	0.3	0.2	0.7	0.5	1.3	0.5	2.2	2	2.3

FF No.: 654

MM0108A: Blockchain

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures : 2 Hrs/week	GD/PPT: 20 Marks
Tutorial : 1 Hrs/week	HA: 10 Marks
	ESE (W): 40Marks

Course Prerequisites:

1. Data Structures
2. Database Management System

Course Objectives:

1: Understand the fundamental concepts, architecture, and evolution of blockchain technology and distributed ledger systems.

2: Explain the role of cryptographic techniques, digital signatures, hash functions, and consensus mechanisms in ensuring blockchain security and trust.

3: Acquire knowledge of Ethereum architecture, smart contracts, Solidity programming, and decentralized application (DApp) development.

4: Explore enterprise blockchain platforms, token standards, and blockchain frameworks for developing scalable decentralized solutions.

5: Analyze blockchain security, privacy-preserving techniques, governance models, and regulatory challenges in blockchain-based systems.

6: Evaluate blockchain applications across various domains and examine emerging trends, innovations, and research opportunities in blockchain technology

Course Relevance:

This course provides a strong foundation in blockchain technology, distributed ledger systems, smart contracts, and decentralized applications. It equips students with the knowledge and practical skills required to develop secure blockchain-based solutions and prepares them for careers in Web3, FinTech, cybersecurity, and enterprise blockchain development.

SECTION-I

Unit 1: Introduction to Blockchain and Distributed Ledger (5 Hours)

Introduction to blockchain technology**,** evolution of blockchain, centralized, decentralized and distributed systems, distributed ledger technology (DLT), blockchain architecture, blockchain components, blocks and transactions, cryptographic hash functions, Merkle trees, public, private and consortium blockchains, blockchain generations (Blockchain 1.0, 2.0, 3.0 and 4.0), advantages and limitations of blockchain, and real-world applications of blockchain technology.

Unit 2: Cryptography and Consensus Mechanisms (5 Hours)

Cryptographic fundamentals, SHA-256 hash function, public key and private key cryptography, digital signatures, blockchain wallets and addresses, consensus mechanisms, Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Proof of Authority (PoA), Practical Byzantine Fault Tolerance (PBFT), mining process, blockchain forks, Byzantine Generals Problem, and comparison of consensus algorithms.

Unit 3: Ethereum and Smart Contracts (5 Hours)

Introduction to Ethereum, Ethereum architecture, Ethereum Virtual Machine (EVM), Ether and gas mechanism, Ethereum accounts and transactions, introduction to Solidity, variables, data types, operators, functions, modifiers, events, arrays, mappings, structures, smart contract development, compilation, deployment and testing using Remix IDE, and introduction to decentralized applications (DApps).

SECTION-II

Unit 4: Decentralized Applications and Enterprise Blockchain (4 Hours)

Decentralized applications (DApps), DApp architecture, Web3 concepts, MetaMask wallet, Web3.js and Ethers.js, token standards (ERC-20 and ERC-721), Non-Fungible Tokens (NFTs), enterprise blockchain, Hyperledger Fabric architecture, permissioned blockchain networks, blockchain-as-a-service (BaaS), and enterprise blockchain use cases.

Unit 5: Blockchain Security and Privacy (4 Hours)

Blockchain security principles, double-spending attack, 51% attack, Sybil attack, eclipse attack, smart contract vulnerabilities, reentrancy attack, blockchain privacy, Zero-Knowledge Proof (ZKP), blockchain governance, legal and regulatory issues, ethical considerations, and best practices for secure blockchain applications.

Unit 6: Blockchain Applications and Emerging Trends (5 Hours)

Blockchain applications in banking and finance, healthcare, supply chain management, digital identity, e-governance, education, Internet of Things (IoT), Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), Central Bank Digital Currency (CBDC), blockchain and Artificial Intelligence (AI), sustainability in blockchain,

future trends, challenges, and research directions.

List of Tutorials:

1. To study and understand the foundational concepts, working mechanisms, advantages, and limitations of Bitcoin, the first successful implementation of blockchain technology.
2. To study how cryptography and consensus mechanisms ensure integrity, security, and decentralization in blockchain technology.
3. Comparative study of various Blockchain platforms (Bitcoin, Ethereum, Hyperledger, Ripple, Corda, R3..etc)
4. Create Wallet and account on any platform and send and receive cryptocurrency.
5. Study the REMIX web based IDE and Solidity Programming.
6. Write a python program to create a simple block in blockchain.
7. Write a Solidity contract using loops and conditionals (e.g., find even numbers).
8. Create a Solidity contract to store and manage student records.
9. Deploying Smart Contracts using Remix IDE
10. Simulate the process of installing, approving, committing, and invoking a chaincode using Hyperledger Fabric Playground (or document it).

List of Course Projects:

1. Blockchain-Based Voting System
2. Academic Certificate Verification System using Blockchain
3. Blockchain-Based Supply Chain Tracking System
4. Decentralized Crowdfunding Platform
5. Land Registration and Property Management System
6. Healthcare Record Management System using Blockchain
7. Blockchain-Based E-Voting with Voter Authentication
8. ERC-20 Cryptocurrency Token Development
9. NFT Marketplace for Digital Assets
10. Blockchain-Based Student Attendance System
11. Decentralized Document Verification and Sharing System
12. Blockchain-Based Charity Donation Tracking System
13. Decentralized Insurance Claim Processing System
14. Blockchain-Based IoT Device Authentication Framework
15. Decentralized Finance (DeFi) Lending and Borrowing Platform

List of Course Seminar Topics: (*if applicable to your course*)

1. Blockchain-Based Secure Academic Credential Verification Framework with AI-Based Forgery Detection
2. Privacy-Preserving Healthcare Data Sharing Using Blockchain and Federated Learning
3. Blockchain-Enabled Smart Supply Chain with Real-Time Traceability and Counterfeit Detection
4. AI-Driven Adaptive Consensus Mechanism for Energy-Efficient Blockchain Networks
5. Blockchain-Based Decentralized Identity Management System for Smart Cities
6. Secure IoT Device Authentication Framework Using Blockchain and Lightweight Cryptography

7. Blockchain-Based Intelligent Land Registration and Property Ownership System
8. Blockchain and Digital Twin Framework for Industrial Asset Integrity Monitoring
9. Decentralized AI Model Marketplace Using Blockchain for Secure Model Sharing
10. Blockchain-Based Carbon Credit Trading Platform for Sustainable Development
11. Trust-Aware Blockchain Framework for Autonomous Vehicle Communication Networks
12. Quantum-Resistant Blockchain Architecture Using Post-Quantum Cryptography
13. Blockchain-Based Secure Electronic Voting System with Biometric Verification
14. Blockchain-Enabled Secure Energy Trading System for Smart Grids and Microgrids
15. Explainable AI and Blockchain Integrated Framework for Transparent Decision Making

Text Books: (As per IEEE format)

[1] I. Bashir, *Mastering Blockchain: A Technical Reference Guide to the Inner Workings of Blockchain, Cryptography, Ethereum, and Smart Contracts*, 4th ed. Birmingham, U.K.: Packt Publishing, 2023.

[2] A. Bahga and V. K. Madisetti, *Blockchain Applications: A Hands-On Approach*. Hyderabad, India: VPT Publications, 2017.

[3] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[4] D. Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 2nd ed. Berkeley, CA, USA: Apress, 2020.

Reference Books: (As per IEEE format)

[1] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.

[2] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton University Press, 2016.

[3] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[4] W. Mougayar, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. Hoboken, NJ, USA: John Wiley & Sons, 2016.

[5] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[6] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation Review*, no. 2, pp. 6–19, Jun. 2016.

Course Outcomes:

The student will be able to –

CO1: Explain the architecture, components, and working principles of blockchain technology and distributed ledger systems.

CO2: Apply cryptographic concepts and compare various consensus mechanisms used in blockchain networks.

CO3: Develop and deploy basic smart contracts using Solidity on the Ethereum blockchain and understand the development of decentralized applications.

CO4: Analyze enterprise blockchain platforms, token standards, and blockchain frameworks for different application scenarios.

CO5: Evaluate blockchain security threats, privacy mechanisms, governance models, and legal considerations for secure blockchain implementations.

CO6: Assess the suitability of blockchain technology for real-world applications in finance, healthcare, supply chain, digital identity, and other emerging domains while identifying future research directions.

Moocs Links and additional reading material: (If any)

1. Introduction to Blockchain Technology and Applications, IIT Kanpur https://nptel.ac.in/courses/106104220
2. Blockchain Architecture Design and Use Cases, IIT Madras https://nptel.ac.in/courses/106105184

Future Courses Mapping:

- Blockchain and Internet of Things (IoT)
- Blockchain for Supply Chain Management
- Blockchain and Artificial Intelligence
- Cloud Computing and Blockchain Integration
- Cybersecurity for Blockchain Systems

Job Mapping: (Optional)

- Full Stack Blockchain Developer
- Research Engineer (Blockchain & Distributed Systems)
- Blockchain Quality Assurance (QA) Engineer
- Blockchain Project Engineer
- Technical Consultant – Blockchain Solutions

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	2												
CO2	2	2	1											
CO3	1	1	2		1			1	2		2	2	2	1
CO4	1	1	2		1				2	1	2	2	2	1
CO5	2	2	1								2	2	1	1
CO6	1	2	1		1	1		1	1	1	1	1		
Average	2	2	1		1	1		1	2	1	2	2	2	1

FF No.: 654

MM0108B: Internet of Things

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures : 2 Hrs/week	GD/PPT: 20 Marks
Practical : 0 Hrs/week	HA: 10 Marks
Tutorial : 1 Hrs/week	ESE: 40 Marks

Course Prerequisites: Computer Network, Programming concepts (Python/C), Digital Electronics and Microprocessor/Microcontrollers, Operating Systems

Course Objectives:

1. Understand the fundamental concepts, architecture, communication models, enabling technologies, protocols, and applications of the Internet of Things (IoT).
2. Analyze and design IoT systems using appropriate platform design methodologies
3. Explain the role of embedded devices, sensors, actuators, networking technologies, and cloud platforms in IoT ecosystems.
4. Evaluate Web of Things (WoT), Cloud of Things (CoT), middleware technologies, and IoT communication protocols for diverse application domains.
5. Design real world applications using IoT.

Course Relevance:

The Internet of Things (IoT) course provides students with a comprehensive understanding of connected devices, communication technologies, IoT architectures, and cloud-based ecosystems that form the foundation of modern smart systems. It introduces the concepts of sensing, data acquisition, communication protocols, and IoT design methodologies, enabling students to understand how physical devices interact with digital platforms.

The course is highly relevant to Artificial Intelligence and Data Science as IoT devices generate vast amounts of real-time data that serve as the primary input for AI and machine learning models. It equips students with the knowledge required to analyze IoT architectures, communication standards, cloud integration, and real-world applications in domains such as smart cities, healthcare, agriculture, industrial automation, transportation, and environmental monitoring. The course also prepares students for advanced studies in Edge Computing, Cyber-Physical Systems, Industrial IoT (IIoT), Smart Systems, AIoT (Artificial Intelligence of Things), Cloud Computing, and Intelligent Automation, while enhancing their understanding of emerging technologies and Industry 4.0.

SECTION-I

Unit 1: Introduction to Internet of Things (5 hours)

Definition and characteristics of IoT, Internet of Things: Vision, Emerging Trends, Economic Significance, TTechnical Building Blocks, Physical design of IoT, Things of IoT, IoT Protocols, Logical design of IoT, IoT functional b blocks, IoT communication models, IoT Communication APIs, IoT enabling technologies, IoT levels

and deployment templates, IoT Issues and Challenges, Applications

Unit 2: IoT Platform Design Methodology (5 Hours)

Purpose and requirement specification, Process specification, Domain model specification, information model specification, Service specifications, IoT level specification, Functional view specification, Operational view specification, Device and component integration, Application development

Unit 3: Pillars of IoT and Physical devices (5 Hours)

Horizontal, verticals and four pillars of IoT, M2M: The internet of devices, RFID: The internet of objects, WSN: The internet of transducer, SCADA: The internet of controllers, DCM: Device, Connect and Manage, Device: Things that talk, Connect: Pervasive Network, Mangae: To create business values. IoT Physical Devices and Endpoints: Basic building blocks of and IoT device, Exemplary device: Raspberry Pi, Raspberry Pi interfaces, Programming Raspberry Pi with Python, Other IoT Devices.

SECTION-II

Unit 4: IoT Protocols (5 Hours)

Protocol Standardization for IoT, Efforts, M2M and WSN Protocols, SCADA and RFID Protocols, Issues with IoT Standardization, Unified Data Standards, Protocols – IEEE 802.15.4, BACNet Protocol, Modbus, KNX, Zigbee Architecture, Network layer, APS layer.

Unit 5: Web of Things and Cloud of Things (4 hours)

Web of Things versus Internet of Things, Two Pillars of the Web, Architecture Standardization for WoT, Platform Middleware for WoT, Unified Multitier WoT Architecture, WoT Portals and Business Intelligence. Cloud of Things: Grid/SOA and Cloud Computing, Cloud Middleware, Cloud Standards – Cloud Providers and Systems, Mobile Cloud Computing, The Cloud of Things Architecture.

Unit 6: IoT Physical Servers, Cloud Offerings and IoT Case Studies (4 hours)

Introduction to Cloud Storage Models, Communication API, WAMP: AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework: Djanjo, Amzon Web Services for IoT, SkyNet IoT Messaging Platform. Case Studies: Home Intrusion Detection, Weather Monitoring System, Air Pollution Monitoring, Smart Irrigation.

List of Tutorials (any 10 tutorials):

Tutorial 1: Introduction to Raspberry Pi, BeagleBoard, Arduino, and Other Microcontrollers
Study the architecture, features, and applications of Raspberry Pi, BeagleBoard, Arduino, and other embedded development boards. Compare their hardware specifications and identify suitable platforms for different embedded and IoT applications.

Tutorial 2: Operating System Installation and Configuration
Learn about different operating systems supported by Raspberry Pi and BeagleBoard. Install, configure, and boot a Linux-based operating system while setting up essential system services.

Tutorial 3: GPIO Programming and LED Interfacing
Understand the concept of General Purpose Input/Output (GPIO) pins and their programming. Interface

LEDs with Raspberry Pi/BeagleBoard and develop basic applications to control LEDs using software.

Tutorial 4: Temperature Sensor Interfacing
Interface a temperature sensor with Raspberry Pi/BeagleBoard and acquire real-time temperature data. Develop an application that monitors the temperature and activates LEDs when a predefined threshold is exceeded.

Tutorial 5: IR Sensor-Based Obstacle Detection
Interface an infrared (IR) sensor to detect the presence of nearby objects. Write an application that indicates obstacle detection using LEDs or other visual indicators.

Tutorial 6: Camera Module Interfacing and Image Capture
Connect a camera module to Raspberry Pi/BeagleBoard and configure it for image acquisition. Develop an application to capture, display, and store images for basic vision-based applications.

Tutorial 7: Zigbee Communication Between Devices
Study the working of Zigbee wireless communication and interface Zigbee modules with Raspberry Pi/BeagleBoard. Develop a simple application to transmit and receive data between two embedded devices.

Tutorial 8: CPU Frequency Scaling and Governors
Understand Linux CPU frequency governors and their role in system performance and power management. Develop an application to monitor and modify the processor operating frequency.

Tutorial 9: Stepper Motor Control
Interface a stepper motor with Raspberry Pi/BeagleBoard using an appropriate driver circuit. Develop an application to control the motor's speed, direction, and movement.

Tutorial 10: Traffic Signal Control System
Design and implement a hardware-simulated traffic signal using LEDs. Develop a control program that follows the standard traffic light sequence with appropriate timing.

Tutorial 11: Lift (Elevator) Control System
Develop a hardware simulation of a lift (elevator) control system using LEDs and switches. Implement floor selection and movement logic to simulate elevator operation.

Tutorial 12: Client–Server Application Development
Develop a server application that runs on Raspberry Pi/BeagleBoard and provides network services. Create client applications that communicate with the server using socket programming over a local network.

Tutorial 13: Cloud-Based IoT Dashboard
Develop a simple cloud-based dashboard for monitoring sensor data from multiple embedded devices. Implement a publisher–subscriber model that enables devices to publish data and users to monitor it remotely.

Tutorial 14: Web-Based Remote LED Control
Develop a web application hosted on Raspberry Pi/BeagleBoard for remote control of connected LEDs. Access the web interface through a browser to monitor and control GPIO devices over the network.

List of Course Projects:

Smart Home Automation System
Home Intrusion Detection and Security System
Weather Monitoring and Forecasting System
Air Pollution Monitoring System
Smart Irrigation System for Agriculture
Smart Street Lighting System
Smart Parking Management System
IoT-Based Water Quality Monitoring System
Smart Energy Meter and Energy Monitoring System
Industrial Equipment Health Monitoring System
IoT-Based Smart Waste Management System
Smart Healthcare and Patient Monitoring System
IoT-Based Fire and Smoke Detection System
Smart Attendance System using RFID
Vehicle Tracking and Fleet Management System
IoT-Based Smart Greenhouse Monitoring System
Cold Storage Monitoring System using IoT Sensors
Smart Water Tank Level Monitoring and Control System
IoT-Based Traffic Density Monitoring System
Smart Campus Monitoring and Management System
IoT-Based Smart Electricity Billing System
Smart Air Quality Monitoring and Alert System
IoT-Based Livestock Monitoring System
Smart Warehouse Monitoring and Asset Tracking System
IoT-Based Flood Detection and Early Warning System
Remote Patient Health Monitoring System using Raspberry Pi
IoT-Based Soil Moisture and Crop Health Monitoring System
IoT-Based Smart Refrigerator Monitoring System
Cloud-Connected Environmental Monitoring System using Raspberry Pi

List of Course Seminar Topics:

Internet of Things (IoT): Vision, Architecture, and Future Trends
Industrial Internet of Things (IIoT) and Industry 4.0
Smart Cities: Applications and Challenges of IoT
Web of Things (WoT) versus Internet of Things (IoT)
Cloud of Things (CoT): Integrating IoT with Cloud Computing
Artificial Intelligence and Machine Learning in IoT Systems
Edge Computing and Fog Computing for IoT Applications
IoT Security, Privacy, and Cybersecurity Challenges
IoT Communication Protocols: MQTT, CoAP, and HTTP
ZigBee, LoRaWAN, and NB-IoT: A Comparative Study
Raspberry Pi as an IoT Development Platform
Wireless Sensor Networks (WSN) in IoT Applications
Machine-to-Machine (M2M) Communication and Its Role in IoT
RFID Technology and Its Applications in IoT

SCADA Systems and Their Integration with IoT
Smart Home Automation Using IoT
IoT-Based Smart Agriculture and Precision Farming
IoT Applications in Healthcare and Remote Patient Monitoring
Smart Transportation and Intelligent Traffic Management Systems
IoT for Environmental and Air Pollution Monitoring
IoT-Based Energy Management and Smart Grids
Digital Twins in IoT Ecosystems
Blockchain Technology for Secure IoT Networks
IoT Middleware Platforms and Service-Oriented Architectures
AWS IoT, Azure IoT, and Google Cloud IoT: A Comparative Analysis
Internet of Medical Things (IoMT) and Healthcare Innovation
Green IoT: Sustainable and Energy-Efficient IoT Systems
Smart Water Management Using IoT Technologies
IoT in Supply Chain and Logistics Management
Emerging Trends in IoT: 5G, 6G, and Beyond
IoT-Based Smart Waste Management Systems
Cloud Storage Models and Data Analytics in IoT
IoT Standardization: IEEE 802.15.4, BACnet, Modbus, and KNX
Smart Campus and Smart Building Solutions Using IoT
Future Research Directions and Challenges in Internet of Things

Text Books: (As per IEEE format)

Arshdeep Bahga, Vijay Madisetti, ―Internet of Things – A hands-on approach‖, Universities Press, ISBN: 0: 0996025510, 13: 978-0996025515

Honbo Zhou, ―The Internet of Things in the Cloud: A Middleware Perspective‖, CRC Press, 2012. ISBN : 9781439892992

Dieter Uckelmann, Mark Harrison, Florian Michahelles, ―Architecting the Internet of Things‖, Springer, 2011. ISBN: 978-3-642-19156-5

Reference Books: (As per IEEE format)

Parikshit N. Mahalle, Poonam N. Railkar, Identity Management for Internet of Things, River Publishers DoI: https://doi.org/10.1201/9781003338505 ebook ISBN 9781003338505

David Easley and Jon Kleinberg, ―Networks, Crowds, and Markets: Reasoning About a Highly Connected World‖, Cambridge University Press, 2010, ISBN:10: 0521195330

Olivier Hersent, Omar Elloumi and David Boswarthick, ―The Internet of Things: Applications to the Smart Grid and Building Automation‖, Wiley, 2012, 9781119958345

Olivier Hersent, David Boswarthick, Omar Elloumi , ―The Internet of Things – Key applications and Protocols‖, Wiley, 2012, ISBN:978-1-119-99435-0

Barrie Sosinsky, ―Cloud Computing Bible‖, Wiley-India, 2010.ISBN : 978-0-470-90356-8 5. Adrian

McEwen, Hakim Cassimally, ―Designing the Internet of Things‖, Wiley, 2014, ISBN: 978-1-118-43063-7

Course Outcomes:

The student will be able to –

CO1: Explain the concepts, characteristics, architecture, communication models, enabling technologies, protocols, challenges, and applications of the Internet of Things (IoT).

CO2: Apply IoT platform design methodologies to develop requirement specifications, domain models, information models, service specifications, and operational views for IoT systems.

CO3: Explain the role of IoT pillars, Raspberry Pi interfaces, sensors, actuators, and Python programming in the development of IoT-based systems.

CO4: Describe IoT communication protocols, standards, and network architectures used in IoT systems.

CO5: Explain the architectures, middleware platforms, standards, cloud computing frameworks, and integration mechanisms of Web of Things (WoT) and Cloud of Things (CoT) for IoT applications.

CO6: Demonstrate the use of cloud platforms, communication APIs, and web frameworks for developing IoT-based applications.

Moocs Links and additional reading material:

Introduction to Internet of Things by Dr. Sudip Misra https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs161

Introduction to Industry 4.0 and Industrial Internet of Things by Dr. Sudip Misra https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs160

Future Courses Mapping:

Cyber-Physical Systems, Ubiquitous Computing, Embedded Systems, Sensor Networks, Edge Computing, Artificial Intelligence for IoT, Data Analytics

Job Mapping:

IoT Support Engineer, Technical Associate, IoT Analyst, IoT System Designer, IoT Solution Engineer, Embedded System Engineer, Embedded Engineer, Hardware Engineer, IoT Device Engineer, Automation Engineer, IoT Application Developer, Full-Stack IoT Developer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	3		1									2	1	
CO2	3	2	2	2	1	1	1		1	1	0	2	1	2
CO3	3	1								1		2	1	1
CO4	3				1					1		2	1	1
CO5	3	1	1		1	1				1		2	1	1
CO6	3	2	2	2	2	1	1		1	1	1	2	1	2
Average	3	1	1	1	1	1	1		1	1	1	2	1	1

FF No.: 654

AI 3205: Design Thinking V

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	0 Hrs/week	ESE: 100 Marks
Tutorial :	1 Hrs/week	

Course Prerequisites:

Basic knowledge of research work, research paper and patent.

Course Objectives:

1. Understand the concepts of design thinking approaches
2. Apply both critical thinking and design thinking in parallel to solve problems
3. Apply some design thinking concepts to their daily work
4. To provide ecosystem for students and faculty for paper publication and patent filing

Course Relevance:

The course is offered in S.Y. and T.Y. B.Tech. to all branches of Engineering.

Contents for Design Thinking:

Structure of The paper Journal List (Top 50 Journals) Selection of the journal

Use of various online journal selection tools Plagiarism checking

Improving contents of the paper Patent drafting

Patent search Filing of patent

Writing answers to reviewer questions Modification in manuscript

Checking of publication draft

Assessment Scheme:

Publication of paper or patent

Course Outcomes:

On completion of the course, learner will be able to– CO1: Understand the importance of doing

Research

CO2: Interpret and distinguish different fundamental terms related to Research

CO3: Apply the methodology of doing research and mode of its publication

CO4: Write a Research Paper based on project work

CO5: Understand Intellectual property rights

CO6: Use the concepts of Ethics in Research

CO-PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	1	2	2	1	3	1	3	2	2	2	1	2	2
CO2	2	2	3	3	2	2	2	3	2	2	1	3	2	3
CO3	2	2	2	2	2	2	2	3	2	2	3	3	2	3
CO4	2	2	2	2	2	2	1	3	2	2	3	1	2	3
CO5	2	2	2	2	2	2	2	3	2	2	3	3	3	2
CO6	2	2	2	2	2	2	2	3	2	2	3	1	3	2
Average	2	1.83	2.16	2.16	1.83	2.16	1.66	3	2	2	2.5	2	2.33	2.5

FF No.: 654

AI3204 - Engineering Design & Innovation VI

Teaching Scheme		Examination Scheme
Credits:	2	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	4 Hrs/week	MSE : 30 Marks
Tutorial :	0 Hrs/week	ESE : 70 Marks

Course Prerequisites:

Problem Based Learning

Course Objectives:

1. To develop critical thinking and problemsolving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, the curriculum can be enriched with semester-long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defence, Education, Smart City, Smart Energy, and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories

of Bloom's Taxonomy

SECTION I

Preamble - The content and process mentioned below is the guideline document for the faculti students to start with. It is not to limit the flexibility of faculty and students; rather they are fre their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory cou contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivit software technologies to solve real world problems using an interdisciplinary approach. The lac form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology(Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intellige Human-Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc

Text Books: (As per IEEE format)

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
3. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro*

Reference Books: (As per IEEE format)

1. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.*
2. *Project management core textbook, second edition, Indian Edition , by Gopalan.*
3. *The Art of Agile Development. By James Shore & Shane Warden.*

Moocs Links and additional reading material:

www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from a societal need point of view

CO2: Choose and compare alternative approaches to select the most feasible one

CO3: Analyse and synthesize the identified problem from a technological perspective

CO4: Select the best possible solution to solve the problem.

CO5: Design & Develop a working model of the proposed solution.

CO6: Testing and validating product performance

Future Courses Mapping:
Major Project

Job Mapping:
Software Engineer. Software Developer, IT Engineer, Research Associate.

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	2			3				2	2			2
CO2				3	2	2		2					2	2
CO3	2		3		3			2	2				2	2
CO4		2				3							2	2
CO5			3				2		2				3	
CO6		2			3				2	3			2	
Average	0.83	1.16	1.33	0.5	1.33	1.33	0.33	0.66	0.5	0.83	0.33		1.83	1.33

FF No.: 654

AI 3010: Deep Learning

Teaching Scheme	Examination Scheme
Credits: 4	ESE: 40 Marks
Lectures : 3Hrs/week	PRACT+CVV: 60 Marks
Practical : 2Hrs/week	
Tutorial : 0 Hrs/week	

Course Prerequisites:

Linear algebra, probability theory and statistics, Digital signal processing, Computer vision

Course Objectives:

1. To present the mathematical, statistical and computational concepts for stable representations of high-dimensional data, such as images,text
2. To introduce NN and techniques to improve network performance
3. To introduce Convolutional Networks
4. To introduce Sequential models of NN
5. To build deep nets with applications to solve real world problem

Course Relevance:

Deep learning is revolutionizing the technology and business world today. It is a subfield of machine learning concerned with algorithms to train computers to perform tasks by exposing neural networks to large amounts of data, its analysis and prediction. It is an incredibly powerful field with capacity to execute feature engineering on its own, and uses multiple neural network layers to extract patterns from the data. Top applications of Deep learning involve self-driving cars, natural language processing, robotics, finance, and healthcare.

SECTION-I

Foundations of neural networks **(08 Hours)**

Foundations of neural networks and deep learning, Logistic regression as a neural network, different activation function, logistic regression cost function, logistic regression gradient descent, Training of neural network by forward and backward propagation

Techniques to improve neural networks **(05Hours)**

Regularization and optimizations, SGD, Momentum, RMSProp, and Adam**,** hyper parameter tuning,

batch normalization, Dropout, data augmentation, Deep learning frameworks, Over fitting concepts, Exploding and Vanishing Gradient Problem

Convolutional Neural Networks **(08 Hours)**

Convolutional Neural Networks, padding, strided convolution, pooling layers, convolutional implementation of sliding windows, ResNet, Transfer Learning

SECTION-II

Sequence Models **(08 Hours)**

Introduction to Sequence Models, Types of Sequence Data, One-to-One, One-to-Many, Many-to-One, Many-to-Many structures, Sequential data types, RNN Architecture, RNN working, LSTM architecture, LSTMs with Keras and Tensorflow, Encoders and Decoders Architecture, Applications of Sequence Models

Transformers **(08 Hours)**

Embeddings, word vectors, self-attention Mechanism, Encoder-decoder structure of Transformer, GPT(Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers), Comparison with traditional sequence models

Object Detection **(05 Hours)**

Introduction to Object Detection, Difference between classification, localization, and detection, Key Components of Detection Models: bounding boxes, anchor boxes, R-CNN: Region proposal + CNN classification, Use-cases

List of Tutorials:

1. Deep learning for Stock Market Clustering
2. Application of Deep Networks in healthcare
3. Credit card fraud detection
4. Classification of skin cancer with deep neural networks
5. ALEXNET
6. VCGNET
7. Accelerating Deep Network Training by Reducing Internal Covariate Shift
8. Deep learning applications for predicting pharmacological properties of drugs
9. GAN (Generative Adversarial Network)
10. Auto encoders
11. LSTM

List of Practical's: (Any Six)

1. Write Python/R code to implement Neural Network from scratch.
2. Write Python/R code to implement Convolutional Neural Network and experiment with different hyper parameters.
3. Write Python/R code to implement Recurrent Neural Network.
4. Write Python/R code to perform Data Augmentation.
5. Write Python/R code to implement LSTM.
6. Write Python/R code to implement GAN.
7. Write Python/R code to implement Sequence Modelling.
8. Write Python/R code to implement Transfer Learning and compare performance of the model with any other model built from scratch. (Use Co-pilot for second code generation)
9. Write Python/R code to implement a self-attention mechanism for any application of your choice.
10. Write Python/R code to implement Deep learning model for Time Series analysis

List of Course Projects:

1. Deep learning for Stock Market Clustering
2. Application of Deep Networks in healthcare
3. Credit card fraud detection
4. Classification of skin cancer with deep neural networks
5. ALEXNET
6. VCGNET
7. Accelerating Deep Network Training by Reducing Internal CovariateShift

List of Course Seminar Topics:

1. Recurrent or Recursive Networks for sequentialModelling?
2. Initializing network weights vs performance
3. Difficulty of training deep feedforward neural networks
4. Hyperparameter tuning: Is there a rule of thumb?
5. Problem of overfitting: How to handle it?

6 Which cost function: Least squared error or binary cross entropy?

7. How to tackle with loss of corner information in CNN
8. Need of hundred classifiers to solve real world classification problem
9. Which optimization: Batch gradient descent of stochastic gradient descent
10. Activation functions: Comparison of trends

Text Books: (As per IEEE format)

1. *Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press,2016.*
2. *Nikhil Buduma, Fundamentals of Deep Learning, O'Reilly, First Edition, ISBN No. 978-14-9192561-4*

Reference Books: (As per IEEE format)

1. *Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd,2009.*
2. *Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.*
3. *SatishKumar,NeuralNetworks:AClassroomApproach,TataMcGraw-HillEducation, 2004*

Moocs Links and additional reading material: www.nptelvideos.in

1. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50

Course Outcomes:

Upon completion of the course, student will be able to –

1. Demonstrate understanding of a logistic regression model, structured as a shallow Neural network
2. Train a deep Neural Network and Apply techniques to improve neural network performance

3. Demonstrate understanding of functionality of all layers in a convolutional neural network
4. Demonstrate Understanding of Recurrent nets and their applications
5. Understand and Apply Architecture of Transformers
6. Illustrate various object detection Techniques.

Future Courses Mapping:

1. NLP

Job Mapping:

Job opportunities that one can get after learning this course

1. Data Scientist
2. Data Engineer

CO - PO Mapping:

	ProgramOutcomes(PO)												PSO			
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0
CO2	3	2	2	0	2	0	0	0	2	0	1	3	2	0	3	0
CO3	3	2	0	2	0	0	0	0	2	0	0	2	2	0	2	2
CO4	2	0	0	2	0	0	0	0	2	0	2	2	0	0	2	3
CO5	2	0	2	0	2	0	0	0	0	0	3	2	2	0	3	0
CO6	2	3	2	2	2	0	0	0	0	0	3	2	2	0	2	2
Average	2.5	1.16	1	1	1.33	0	0	0	1	0	1.5	2.16	1.33	0	2	1.16

FF No.: 654

AI 3011: Complexity and Algorithms

Teaching Scheme		Examination Scheme
Credits:	3	CP: 30 Marks
Lectures :	2 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	Lab Assessment: 10 Marks
Tutorial :	-- Hrs/week	ESE (w) : 40 Marks

Course Prerequisites:

Basic course on Programming, Data structures, Discrete structures

Course Objectives:

1. Formulate a given computational problem in an abstract and mathematically precise manner.
2. Choose a suitable paradigm to design algorithms for given computational problems.
3. Understand asymptotic notations and apply suitable mathematical techniques to find algorithms' asymptotic time and space complexities.
4. Understand the notion of NP-hardness and NP-completeness and the relationship with the intractability of decision problems.
5. Apply randomized, approximation algorithms for given computational problems.

Course Relevance:

This is an important course for AI-DS Engineering. It develops the algorithmic thinking capability of students. Designing algorithms using suitable paradigms and analyzing the algorithms for computational problems has a high relevance in all domains of IT(equally in Industry as well as research). Once the student gains expertise in Algorithm design and in general gains the ability of algorithmic thinking, it facilitates systematic study of any other domain (in IT or otherwise) which demands logical thinking. This course is also relevant for students who want to pursue research careers in theory of computing, computational complexity theory, and advanced algorithmic research.

SECTION-I

Unit 1: Introduction (5 Hours)

Introduction: Algorithm analysis,Time complexity: Growth of Function: Asymptotic notation, Standard notations and ɔmmon functions Complexity analysis-Time and space trade offs in algorithms(Big Oh, small oh, Big Omega, Theta ɒtations). Best case, average case, and worstcase time and space complexity of algorithms., Recurrences. Methods for nding complexity of recursive algorithms, Analysis of selection and Insertion sort. (6 Hours)

Divide and Conquer: Binary search, Quick sort, Merge Sort: Analysis of all algorithms. (3 Hours)

Greedy Design Strategies: Elements of Greedy Strategy, Analysis and correctness proof of Minimum cost spanning tree gorithms and shortest path algorithms, , Job Sequencing with deadline, Fractional Knapsack, Huffman coding (5 Hours)

SECTION-II

Dynamic Programming: General strategy, Dynamic programming for optimization problems, Principle of optimality, Multistage graphs, Optimal Binary Search Tree, Single source shortest path, All-pairs shortest path, 0/1 Knapsack problem, Traveling Salesperson problem. (5 Hours)

Backtracking: General method, Recursive backtracking algorithm, Iterative backtracking method. N-Queen problem, Sum of subsets, Graph colouring, Hamiltonian Cycle (3 Hours)

Computational Complexity: Deterministic and Non-Deterministic algorithms (1 Hour)

Complexity Classes: P, NP and NP -Completeness Problems, Satisfiability problem, Proofs for NP Complete (3 Hours)

Introduction to Randomized and Approximation algorithms: Introduction to randomness in computation, Las-Vegas and Monte-Carlo algorithms (2 Hours)

List of Practical's: (Any ….)

1. Basics: Find out Big - Oh and Big – Omega of the function. Take necessary data like degree of the function, coefficients, etc… .
2. Some Basic Algorithms: Write an algorithm and find the efficiency of the same for following problems:

   a. Finding Factorial – Iterative Approach

   b. Finding Factorial – Recursive Approach

c. Printing Fibonacci Series – Iterative Approach

d. Printing Fibonacci Series – Recursive Approach

3. Basic Sorting and Searching Techniques: Assignment based on analysis of quick sort(deterministic and randomized variant).

4. Divide and Conquer Approach: Assignment based on Divide and Conquer Strategy(e.g. majority element search, finding kth rank element in an array).

5. Divide and Conquer Approach: Assignment based on Divide and Conquer strategy (e.g. efficient algorithm for Josephus problem using recurrence relations, fast modular exponentiation).

6. Dynamic Programming: Assignment based on Dynamic Programming strategy(e.g., All pair shortest path, Traveling Salesperson problem).

7. Greedy Approach: Design an algorithm and implement a program to solve:

a. Making Change Problem

b. Knapsack Problem

c. Huffman encoding

8. Backtracking: Assignment based on Backtracking(e.g. graph coloring-queen problem).

9. Randomized Algorithms: Assignment based on Las-Vegas and Monte-Carlo algorithm for majority element search.

10. Approximation Algorithms: Assignment based on factor-2 approximation algorithm for metric - TSP

List of Tutorials:

1. Asymptotic Notation: Big-O, Big-Ω, Big-Θ

2. Recurrence Relations and Master Theorem

3. Divide and Conquer Strategy

4. Greedy Algorithms

5. Dynamic Programming: Concepts and Patterns

6. Backtracking and Problem Solving

7. Minimum Spanning Tree: Kruskal and Prim

8. Topological Sorting and Cycle Detection in Graphs

9. String Matching Algorithms

10. NP-Completeness and Reductions

11. Approximation Algorithms

12. Randomized Algorithms: Basics and Examples

13. Complexity Classes: P, NP, NP-Complete, NP-Hard

List of Course Projects:

1. Visualizing and Comparing Sorting Algorithms :Implement and visualize Bubble, Merge, Quick, and Heap sort with time/space benchmarks.

2. Time Complexity Analyzer Tool :Build a tool to estimate time complexity of Python/C++ functions using empirical testing

3. Recursive vs Iterative Algorithms: Performance Case Study :Compare performance for Fibonacci, Tree

Traversals, Factorial, etc.

4. Solving NP-Complete Problems with Heuristics :Pick a problem like Knapsack, Sudoku, or Graph Coloring and solve using greedy, backtracking, and approximation.

5. Approximate Algorithms for the Traveling Salesman Problem (TSP) :Implement Nearest Neighbor, Minimum Spanning Tree heuristic, or Genetic Algorithms for TSP.

6. Simulation of a Job Scheduler with Complexity Analysis :Create a simulated CPU job scheduler and analyze its algorithmic performance.

7. Time-Space Tradeoffs in Algorithms : Analyze real-world examples (e.g., memoization vs recomputation) and measure differences.

8. Randomized Algorithms in Practice : Implement and compare randomized QuickSort, Min-Cut, or hash-based algorithms.

List of Course Seminar Topics: (*if applicable to your course*)

1. Complexity classes
2. Time and Space Complexity: How We Measure Efficiency
3. Divide and Conquer Vs Dynamic Programming
4. Greedy vs Dynamic Programming: When and Why They Work
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Applications of Recursion and Backtracking
8. P vs NP Problem: Current Perspectives
9. Graph Algorithms in Everyday Life
10. NP-Completeness and Reductions: Techniques and Case Studies

Text Books: (As per IEEE format)

1.Horowitz and Sahani, Fundamentals of computer Algorithms, Galgotia, ISBN 81-7371-612-9.

2.Thomas H Cormen and Charles E.L Leiserson, Introduction to Algorithm, PHI, ISBN:81-203- 2141-3.

Reference Books: (As per IEEE format)

1. SanjoyDasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGrawHillEdition.

2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI.

3. John Kleinberg, Eva Tardos, "Algorithm Design", Pearson.

4. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design", Wiley Publication

Course Outcomes:

The student will be able to –

1. Understand the basic notation for analyzing the performance of the algorithms.

2. Apply appropriate algorithmic paradigms to design efficient algorithms for computational problems

3. Apply suitable mathematical techniques to analyze the asymptotic complexity of the algorithm for more complex computational problems.

4. Understand the significance of NP-completeness of some decision problems and its relationship with the tractability of the decision problems.

5. Understand the significance of randomness, and approximability in computation and design randomized and approximation algorithms for suitable problems

6.Incorporate appropriate data structures, and algorithmic paradigms to craft innovative scientific solutions for complex computing problems

Moocs Links and additional reading material: (If any)

1. www.nptelvideos.in

Future Courses Mapping:

1. Advanced Algorithms

2. Graph Theory and Network Algorithms

3. Parallel and Distributed Algorithms

Job Mapping: (Optional)

1. Software Engineer / Software Developer

2. Data Scientist / Machine Learning Engineer

3. Software Performance Engineer

4. Game Developer (AI/Pathfinding Specialist)

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	3	2					1	2		3	3	
CO2	3	3	3	2	1	1	1				2	2		2
CO3	2	2	3	3	2					2		2		
CO4	3	2										2		
CO5	3	2	3			2			3				1	3
CO6	0	3	2	3	3		3			3				2
Average	2.8	2.3	2.8	2.2	2		2		2	2.3	2	2.25	2	2.3

FF No.: 654

AI 3207A: Quantum Computing

Teaching Scheme		Examination Scheme
Credits:	4	CP: 40 Marks
Lectures :	3 Hrs/week	CVV:20 Marks
Practical :	2 Hrs/week	GD/PPT: 20 Marks
Tutorial :	NA	HA : 20 Marks

Course Prerequisites:

1. Engineering Mathematics
2. Data Structures and Algorithm
3. Python Programming

Course Objectives:

1. To understand basics of quantum computing
2. To understand mathematics required for quantum computing
3. To understand building blocks of quantum computing and design algorithms
4. To understand quantum hardware principles and tools for quantum computing.

Course Relevance: Quantum Computing is an emerging technology with the potential to solve complex computational problems beyond the capabilities of classical computers. This course introduces the fundamental principles of quantum mechanics, quantum algorithms, quantum programming, and quantum hardware. Students gain hands-on experience with Qiskit and IBM Quantum platforms to design and simulate quantum circuits. The course also explores applications in artificial intelligence, cryptography, optimization, finance, healthcare, and cybersecurity, preparing students for advanced studies and careers in quantum technologies, quantum software development, and quantum AI.

SECTION-I

Unit 1: Introduction to Quantum Computing (5 Hours)

Motivation for studying Quantum Computing , Origin of Quantum Computing , Quantum Computer vs. lassical Computer , Introduction to Quantum mechanics , Overview of major concepts in Quantum omputing.

Overview of Quantum Logic gates and Circuit, Qubits and multi-qubits states , Bloch Sphere representation, Quantum Superposition, Quantum Entanglement, Major players in the industry (IBM, Microsoft, Rigetti, D-Wave etc.)

Unit 2: Mathematical Foundations for Quantum Computing (4 Hours)

Matrix Algebra : basis vectors and orthogonality , inner product and Hilbert spaces , matrices and tensors , unitary operators and projectors , Dirac notation , Eigen values and Eigen vectors.

Unit 3: Building Blocks for Quantum Program (5 Hours)

Architecture of a Quantum Computing platform, Quantum Circuit Model, Quantum Gates, Single- and Multi-Qubit Systems, Entanglement, Measurement, No-Cloning Theorem, Teleportation.

Programming model for a Quantum Computing Program, Steps performed on classical computer and quantum computer, Moving data between bits and qubits, Introduction to IBM Quantum Platform and Qiskit, quantum circuit creation, simulation, execution, and visualization.

SECTION-II

Unit 4: Quantum Algorithms (4 Hours)

Quantum Algorithms- Deutsch‘s Algorithm , Deutsch -Jozsa Algorithm, Grover‘s Algorithm, Shor‘s Algorithm.

Unit 5: Quantum Fourier Transform and Applications (6 Hours)

Quantum Fourier Transform, Phase estimation, period finding, order finding, factoring application, General applications of Quantum Fourier transform discrete algorithms, Quantum error correction using- repetition codes, 3 qubit codes, Shor‘s 9 qubit error correction Code.

Unit 6: Quantum Machine Learning and Emerging Applications (4 Hours)

Introduction to Quantum Machine Learning, Variational Quantum Circuits, Quantum Kernels, Quantum Neural Networks, Hybrid Quantum–Classical Learning, Quantum Natural Language Understanding, Application Domains for Quantum Machine Learning: Chemistry/Material Science, AI, Healthcare, Finance, Cybersecurity, Robotics

List of Practical's:

1. Understand and simulate basic quantum states and gates and build simple quantum circuits with single and two-qubit gates
   1. Install and set up Qiskit
   2. Create a quantum circuit with a single qubit
   3. Apply X, H, and Z gates
   4. Measure the qubit state
   5. Visualize results using Bloch sphere and statevector simulator

2. Analyze simple states of superposition and the effect of doing the measurement in different basis states .
   1. Apply Hadamard gate to create superposition
   2. Measure the qubit multiple times
   3. Analyze the output distribution using histograms
   4. Compare simulator vs real quantum hardware results

3. Learn how to use the IBM infrastructure to write quantum programs in QASM (Quantum Assembly) language.

4. Analyze quantum circuits with entanglement
   1. Build a circuit to create a Bell state (e.g., $|\Phi^+\rangle$)
   2. Use CNOT and Hadamard gates
   3. Measure both qubits
   4. Confirm correlation using histogram results

5. Analyze the effectiveness of a simple error correction scheme.

6. Implement quantum programs in the NISQ model of computing.

List of Tutorials: –

1. Introduction to Quantum Computing and Qubit Representation
2. Linear Algebra for Quantum Computing
3. Quantum Gates and Circuit Design
4. Quantum Superposition and Entanglement
5. Quantum Programming with Qiskit
6. Analysis of Fundamental Quantum Algorithms
7. Quantum Fourier Transform and Phase Estimation
8. Quantum Error Detection and Error Correction
9. Quantum Hardware and IBM Quantum Experience
10. Quantum Machine Learning Applications

List of Course Projects: –

1. Quantum Secure Voting System Simulator

   Develop a secure electronic voting prototype using quantum cryptography concepts and quantum key distribution to demonstrate secure communication.

2. Quantum Random Number Generator (QRNG) for Secure Applications

   Generate truly random numbers using quantum superposition and compare randomness with classical pseudo-random generators.

3. Quantum Password Authentication using BB84 Protocol

   Simulate a secure authentication mechanism using BB84 Quantum Key Distribution for encrypted password transmission.

4. Quantum Image Edge Detection

   Implement a basic quantum image processing algorithm for detecting edges in grayscale images using quantum circuits.

5. Quantum Traffic Signal Optimization

   Model traffic intersections as optimization problems and solve them using Grover-inspired search techniques.

6. Quantum Recommendation System Prototype

   Develop a simple movie or product recommendation model enhanced with quantum-inspired search and similarity computation.

7. Quantum Portfolio Optimization

   Simulate investment portfolio optimization using quantum optimization algorithms and compare with classical methods.

8. Quantum Sudoku Solver

   Implement a quantum search-based Sudoku solver demonstrating the efficiency of Grover's Algorithm.

9. Quantum Maze Solver

   Solve maze pathfinding problems using quantum search concepts and compare execution with classical search algorithms.

10. Quantum Supply Chain Optimization

    Model warehouse routing or delivery optimization problems using quantum optimization principles.

11. Quantum Superdense Coding Demonstration

    Implement superdense coding to transmit two classical bits using one qubit and analyze

communication efficiency.

12. Quantum Algorithm Performance Analyzer

    Compare Deutsch-Jozsa, Bernstein-Vazirani, Grover, and Shor algorithms with classical approaches in terms of complexity and execution.

13. Quantum Machine Learning for Iris Classification

    Implement a simple Quantum Support Vector Machine (QSVM) or Variational Quantum Classifier (VQC) for the Iris dataset.

14. Quantum Fraud Detection Prototype

    Apply quantum machine learning techniques to detect fraudulent financial transactions using benchmark datasets.

List of Course Seminar Topics: (*if applicable to your course*)

1. Evolution of Quantum Computing: From Classical to Quantum Computers
2. Qubits, Superposition, and Quantum Measurement
3. Quantum Entanglement and Its Applications
4. Differences Between Classical and Quantum Computing
5. Quantum Computer Architecture and Components
6. Quantum Gates and Quantum Circuits
7. Quantum Programming Languages: Qiskit, Cirq, and PennyLane
8. Quantum Circuit Design and Simulation
9. Linear Algebra in Quantum Computing
10. Complex Numbers, Vector Spaces, and Dirac Notation
11. Superconducting Quantum Processors
12. Trapped-Ion Quantum Computers
13. Photonic Quantum Computing
14. Quantum Error Correction and Fault-Tolerant Quantum Computing
15. Grover's Search Algorithm and Its Applications
16. Shor's Factorization Algorithm
17. Quantum Fourier Transform and Phase Estimation
18. Quantum Machine Learning Algorithms

Text Books: (As per IEEE format)

1. Quantum Computation and Quantum Information M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed. Cambridge, U.K.: Cambridge University Press, 2010.
2. Quantum Computing Explained D. McMahon, *Quantum Computing Explained*. Hoboken, NJ, USA: John Wiley & Sons, 2008.
3. Qiskit Textbook (Archived)
4. Practical Quantum Computing for Developers V. Silva, *Practical Quantum Computing for*

Developers. Berkeley, CA, USA: Apress, 2018.

Reference Books: (As per IEEE format)

1. B. Zygelman, A First Introduction to Quantum Computing and Information. Cham, Switzerland: Springer, 2018.
2. Introduction to Spintronics S. Bandyopadhyay and M. Cahay, Introduction to Spintronics. Boca Raton, FL, USA: CRC Press, 2008.
3. The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies L. Jaeger, The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies. Cham, Switzerland: Springer, 2018.
4. Quantum Error Correction Codes G. G. La Guardia, Quantum Error Correction Codes. Cham, Switzerland: Springer, 2021.

Course Outcomes:

The student will be able to –

1. **Understand** basic concepts of quantum computing. **L2 - Understand**
2. **Illustrate** building blocks of quantum computing through architecture and programming models. **L2 - Understand**
3. **Appraise** various mathematical models required for quantum computing. **L5 Evaluation**
4. **Apply** quantum hardware building principles to explain the operation of quantum computing systems. **L3- Apply**
5. **Analyze** various quantum algorithms based on their design, computational complexity, and application domains. **L4 - Analyze**
6. **Apply** quantum computing tools to design and execute basic quantum programs and analyze their outputs. **L3- Apply**

Moocs Links and additional reading material: (If any)

1. https://onlinecourses.nptel.ac.in/noc21_cs103/preview
2. https://www.coursera.org/courses?query=quantum%20computing
3. https://www.cl.cam.ac.uk/teaching/1617/QuantComp/

Future Courses Mapping:

Quantum Algorithm Engineer, Quantum AI Engineer, Quantum Computing Research Scientist, Quantum Software Development, Quantum AI Applications, Quantum Cloud Computing

Job Mapping: (Optional)

Quantum Software Engineer, Quantum Application Developer, Quantum Cloud Engineer

CO-PO

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO 12**	**PSO1**	**PSO2**
CO1	3	1	-	-	-	-	-	-	-	-	-	1	2	1
CO2	3	2	1	-	2	-	-	-	-	-	-	1	2	2
CO3	3	3	-	2	1	-	-	-	-	-	-	2	3	2
CO4	3	2	2	1	2	-	-	-	-	-	-	1	2	2
CO5	3	3	2	2	2	-	-	-	-	1	-	2	3	3
CO6	2	2	2	2	3	-	-	-	1	1	-	2	3	3
Average	2.83	2.17	1.75	1.75	2.00	-	-	-	1.0	1.0	-	1.50	2.50	2.17

FF No.: 654

AI 3207B: Federated Learning

Teaching Scheme	Examination Scheme
Credits: 4	CP: 40 Marks
Lectures : 3 Hrs/week	CVV: 20 Marks
Practical : 2 Hrs/week	GD/PPT : 20 Marks
Tutorial : 0 Hrs/week	HA : 20 Marks

Course Prerequisites: Machine Learning, Distributed systems, Privacy and security.

Course Objectives:

CO1 (Unit 1):
To provide students with a comprehensive understanding of the fundamental concepts, architecture, and applications of Federated Learning.

CO2 (Unit 2):
To equip students with the knowledge of machine learning techniques and personalization methods used in Federated Learning systems.

CO3 (Unit 3):
To enable students to analyze and apply optimization techniques for improving the performance of Federated Learning under heterogeneous and non-IID data settings.

CO4 (Unit 4):
To develop students' understanding of distributed Federated Learning paradigms, including horizontal, vertical, and transfer learning frameworks.

CO5 (Unit 5):
To familiarize students with privacy-preserving mechanisms and security techniques for protecting Federated Learning systems against data leakage and adversarial attacks.

CO6 (Unit 6):
To enable students to design and implement Federated Learning solutions for real-world applications in healthcare, finance, IoT, and other distributed intelligent systems.

Course Relevance:

Federated Learning is an emerging paradigm in Artificial Intelligence that enables collaborative model training across distributed devices while preserving data privacy and security. This course builds upon students' knowledge of machine learning, distributed computing, and data analytics to address real-world challenges associated with decentralized data. It equips students with the skills to design, optimize, and deploy privacy-preserving AI solutions for applications in healthcare, finance, IoT, autonomous systems, and smart cities. The course prepares students for advanced research and industry roles in trustworthy AI, edge intelligence, distributed machine learning, and privacy-aware data science.

SECTION-I

Unit 1 Introduction to Federated Learning (7 Hrs)

Overview of Federated Learning: Definition, History, and Applications Concepts and Terminology, Machine Learning perspective, Security & Privacy, Federated Learning vs Centralized Learning: Comparison and Contrast

Case Studies: FL for Google Assistant 'Hey Google' or Alexa

Unit 2 Federated Learning as a Machine Learning Problem (7 Hrs)

Tree based models in FL Semantic vectorization : Text and Graph based models, Personalization, Personalized and Robust FL with Fed+,MapReduce algorithm
Case Studies: DFedForest : Decentralized Federated Forest

Unit 3 Optimization Techniques (7 Hrs)

Federated learning with Non-IID Data: Heterogeneity, Stratification and Local Updated rules. Advanced optimization techniques: Adaptive Learning rate, Momentum and Weight decay, Meta learning for FL: Model Selection and hyper parameter tuning.
Case Studies: FL for Autonomous Vehicles

SECTION-II

Unit 4 Distributed Machine Learning (7 Hrs)

Horizontal FL: Definition, Architecture, Federated Averaging algorithm, Improvement of FedAvg algorithm
Vertical FL: Definition, Architecture, VFL algorithms: Secure Federated Linear Regression, Secure Federated Tree Boosting,

Federated Transfer Learning: Framework, Homomorphic encryption, FTL training process, FTL prediction process, Security analysis, Secret sharing based FTL.

Case Studies: Movie recommendation for user's past browsing behaviour using Vertical FL.

Unit 5 Privacy & Security (7 Hrs)

Protecting against Data Leakage in FL, Private parameter aggregation for FL, Data leakage in FL, Dealing with Byzantine threats to Neural Networks.

Case Studies: Federated Learning for Collaborative Financial Crimes Detection

Unit 6 Applications (7 Hrs)

Federated Learning for Healthcare: Privacy preserving Diagnosis and Treatment planning

Finance: Fraud detection and Risk Assessment

IoT: Edge Computing and Distributed Sensing.

Case studies: Case study for COVID-19

List of Practical's: (Any ….)

1. **Install and configure a Federated Learning framework** (e.g., IBM Watson Studio, TensorFlow Federated, Flower, or FedML) and create a basic Federated Learning environment.

2. **Perform data preprocessing and partitioning** of a dataset into multiple clients to simulate decentralized data distribution (IID and Non-IID scenarios).

3. **Implement local model training** on multiple participating clients and transmit model updates to the central server.

4. **Implement the Federated Averaging (FedAvg) algorithm** to aggregate local model updates at the server and distribute the updated global model to participating clients.

5. **Analyze model convergence and performance** by executing multiple communication rounds and evaluating accuracy, loss, and communication efficiency.

6. **Implement privacy-preserving Federated Learning** using Differential Privacy techniques and compare its performance with standard Federated Learning.

List of Course Projects:

1. Privacy-Preserving Handwritten Digit Classification using Federated Learning
   - Train an MNIST classifier across multiple clients using FedAvg.
2. Federated Sentiment Analysis
   - Develop a distributed sentiment analysis model where clients retain local text data.
3. Federated Healthcare Disease Prediction
   - Train a disease prediction model collaboratively using distributed patient datasets while preserving privacy.
4. Credit Card Fraud Detection using Federated Learning
   - Build a privacy-preserving fraud detection model across multiple banking clients.
5. Federated Learning for Smart Home IoT Devices
   - Implement collaborative anomaly detection using IoT sensor data collected from multiple homes.
6. Federated Plant Disease Classification

- Train an image classification model using decentralized agricultural image datasets.

7. Federated Recommendation System
   - Develop a movie or product recommendation system using user data stored locally on different clients.
8. Non-IID Data Analysis in Federated Learning
   - Compare model performance under IID and Non-IID client data distributions.
9. Secure Federated Learning using Differential Privacy
   - Implement differential privacy and evaluate the trade-off between privacy and model accuracy.
10. Performance Comparison of Federated Learning Frameworks
    - Compare TensorFlow Federated, Flower, and FedML in terms of communication cost, convergence speed, and model accuracy.
11. Federated Image Classification using CIFAR-10
    - Train a CNN collaboratively on decentralized image datasets and evaluate global model performance.
12. Communication-Efficient Federated Learning
    - Study the effect of varying communication rounds, client participation, and aggregation frequency on training efficiency.

List of Course Seminar Topics: (*if applicable to your course*)

1. Evolution of Federated Learning and its Future Directions
2. Federated Learning vs. Centralized Machine Learning
3. Horizontal, Vertical, and Federated Transfer Learning
4. Federated Averaging (FedAvg) and Advanced Aggregation Algorithms
5. Personalized Federated Learning
6. Federated Learning with Non-IID Data
7. Communication-Efficient Federated Learning
8. Privacy-Preserving Techniques in Federated Learning
9. Differential Privacy in Federated Learning
10. Secure Aggregation and Homomorphic Encryption
11. Byzantine Attacks and Defense Mechanisms in Federated Learning
12. Blockchain-enabled Federated Learning

13. Federated Learning for Healthcare Applications
14. Federated Learning in Smart Cities and IoT
15. Federated Learning for Autonomous Vehicles
16. Federated Learning for Financial Fraud Detection
17. Federated Learning for Recommender Systems
18. Edge AI and Federated Edge Learning
19. Federated Learning for Large Language Models (LLMs)
20. Federated Learning in 6G Networks and Edge Computing
21. Explainable Artificial Intelligence (XAI) in Federated Learning
22. Green Federated Learning: Energy-Efficient Distributed AI
23. Recent Industrial Applications of Federated Learning (Google, Apple, NVIDIA, IBM, Microsoft)
24. Open-Source Federated Learning Frameworks: TensorFlow Federated, Flower, FedML, PySyft
25. Challenges and Future Research Directions in Federated Learning

These seminar topics are aligned with syllabus units and expose students to **current research and industrial trends**, making them suitable for **TY AI&DS** students.

Text Books: (As per IEEE format)

1. H. Ludwig and N. Baracaldo, Eds., Federated Learning: A Comprehensive Overview of Methods and Applications, 1st ed. Cham, Switzerland: Springer Nature Switzerland AG, 2022. doi: 10.1007/978-3-030-96896-0.

2. Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, Federated Learning. San Rafael, CA, USA: Morgan & Claypool Publishers, 2020, vol. 13, no. 3, Synthesis Lectures on Artificial Intelligence and Machine Learning. doi: 10.2200/S00960ED2V01Y201910AIM043.

Reference Books: (As per IEEE format)

1. K. Nakayama and G. Jeno, Federated Learning with Python: Design and Implement a Federated Learning System and Develop Applications Using Existing Frameworks. Birmingham, U.K.: Packt Publishing Ltd., 2022. ISBN: 978-1-80324-710-6

2. E. Glanz and N. Fallen, What Is Federated Learning? Sebastopol, CA, USA: O'Reilly Media, Inc., Oct. 2021. ISBN: 978-1-098-10725-3.

Course Outcomes:

The student will be able to –

1.Describe the key concepts and architecture of Federated Learning

2. Apply different methods to develop federated learning systems.

3. Apply optimization techniques in FL.

4. Construct and scale a simple federated system.

5. Evaluate privacy and security concerns in Federated Learning and implement privacy-preserving techniques.

6. Implement Federated Learning in real-world applications

Moocs Links and additional reading material: (If any)

https://research.aimultiple.com/federated-learning/

https://towardsdatascience.com

https://www.youtube.com/watch?v=I7j5x18igp4

https://www.youtube.com/watch?v=YsFVdDAgkcU

https://www.nature.com/articles/s41746-021-00431-6

Future Courses Mapping:

1. Advanced Machine Learning
2. Deep Learning
3. Distributed Artificial Intelligence
4. Edge AI and Edge Computing
5. Privacy-Preserving Machine Learning
6. Trustworthy and Explainable AI
7. Cloud Computing and Distributed Systems
8. Internet of Things (IoT) and Edge Intelligence
9. Big Data Analytics
10. Reinforcement Learning
11. Artificial Intelligence for Healthcare
12. AI Security and Privacy
13. Blockchain for AI Applications
14. Research Project / Capstone Project
15. M.Tech./Ph.D. courses in Artificial Intelligence, Machine Learning, Distributed Systems, and Data Science

Job Mapping: (Optional)

- AI/ML Engineer
- Data Scientist
- Machine Learning Engineer
- Deep Learning Engineer

- Federated Learning Engineer
- Edge AI Engineer
- AI Research Engineer
- Computer Vision Engineer
- NLP Engineer
- MLOps Engineer
- Cloud AI Engineer
- IoT and Edge Computing Engineer
- AI Security and Privacy Engineer
- Big Data Engineer
- Research Associate (Artificial Intelligence)

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	3	2	-	-	2	-	-	-	-	-	-	-	2	1
CO2	3	3	2	-	3	-	-	-	-	-	-	-	3	2
CO3	2	3	2	2	3	-	-	-	-	-	-	-	3	3
CO4	3	2	3	-	3	-	-	-	2	-	-	-	3	3
CO5	2	3	2	2	3	2	-	3	-	-	-	-	3	2
CO6	2	2	3	2	3	2	2	2	2	2	-	2	3	3
Average	2.5	2.5	2	1	2.83	0.67	0.33	0.83	0.67	0.33	0	0.33	2.83	2.33

FF No.: 654

AI3207C: Software Design and Methodologies

Teaching Scheme	Examination Scheme
Credits: 4	CP: 40 Marks
Lectures : 3 Hrs/week	CVV: 20 Marks
Practical : 2 Hrs/week	GD/PPT:20 Marks
Tutorial : 0 Hrs/week	HA : 20 Marks

Course Prerequisites:

Proficient of programming in a high-level, object-oriented language, familiarity with data structures and algorithms

Course Objectives:

1. Understanding object-oriented analysis and design.
2. Learn different software process models and principles and practices
3. Practicing UML to model OO systems
4. Familiarity with current models and standards for design.
5. Exposure to organizational issues in software design.
6. The skill to analyze problems critically, leveraging both theoretical and technical knowledge to devise solutions and systems

Course Relevance:

Software Architecture

SECTION-I

Overview of Software Engineering :

Software Process Framework, Process Patterns, Process Models: Code-and-Fix, Waterfall Model, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Agile process, Software Engineering Principles and Practices. **(5 Hours)**

Software Modeling : Introduction to Software Modeling, Advantages of modeling, Principles of modeling. **(3 Hours)**

Evolution of Software Modeling and Design Methods :

Object oriented analysis and design methods, Concurrent, Distributed Design Methods and Real-Time Design Methods, Model Driven Architecture (MDA), 4+1 Architecture, Introduction to UML, UML building Blocks, COMET Use Case–Based Software Life Cycle. **(5 Hours)**

Requirement Study :

Requirement Analysis, SRS design, Requirements Modeling. Use Case: Actor and Use case identification, Use case relationship (Include, Extend, Use case Generalization, Actor Generalization), Use case template **(4 Hours)**

Study of classes (analysis level and design level classes) **(2 Hour)**

Methods for identification of classes :

RUP (Rational Unified Process), CRC (Class, Responsibilities and Collaboration), Use of Noun Verb analysis (for identifying entity classes, controller classes and boundary classes). **(4 Hours)**

SECTION-II

Class Diagram :

Relationship between classes, Generalization/Specialization Hierarchy, Composition and Aggregation Hierarchies, Associations Classes, Constraints. Object diagram, Package diagram, Component diagram, Composite Structure diagram, Deployment Diagram. **(4 Hours)**

Activity Diagram :

Different Types of nodes, Control flow, Activity Partition, Exception handler, Interruptible activity region, Input and output parameters, Pins. **(4 Hours)**

Interaction Diagram :

Sequence diagram, Interaction Overview diagram, State machine diagram, Advanced State Machine diagram, Communication diagram, Timing diagram. **(4 Hours)**

Architecture in the Life Cycle :

Architectural styles, Architecture in Agile Projects, Architecture and Requirements, Designing an Architecture. **(4 Hours)**

Design Patterns :

Introduction, Different approaches to select Design Patterns, Creational patterns, Structural Patterns, Behavioral Patterns **(3 Hours)**

List of Practical's:

1. To study modeling methodologies and identify their applicability to various categories of projects.
2. To understand Requirement Elicitation Techniques and recognize types of requirements while preparing System Requirement Specification.
3. To study MDD/MDA and identify the importance of Model Transformation.
4. To study the various types of AI tools for designing UML2.0. diagrams

5. To identify System Scope, Actors, Use Cases, Use Case structuring for a given problem and perform Use Case narration in template form with normal/alternate flows.
6. ToidentifyEntity,Control,Boundaryobjectsandtraceobjectinteractionsforscenariosfrom use cases.
7. Prepare a state chart diagram for a given object scenario.
8. TopreparedetailedActivitydiagramwithnotationalcompliancetoUML2.0indicatingclearuse of pins, fork-join, synchronization,datastores
9. To prepare Class diagrams for a defined problem with relationships, associations, hierarchies, interfaces, roles and multiplicity indicators.
10. To prepare a Component and Deployment diagram for a defined problem.

List of Tutorials:

1. Goals of software engineering
2. Software process models, life cycle models
3. Process improvement, Capability Maturity Model
4. Unified Modeling Language(UML)
5. Design patterns
6. Frameworks, software product lines
7. Software architecture
8. Software measurements and metrics
9. Software estimation methods
10. Static and dynamic analysis
11. Version control, configuration management
12. Software quality, verification and validation, software testing

List of Course Projects:

1. Weather prediction system management
2. Agricultural water management system
3. ERP system
4. Hospital Management

5. Railway Reservation
6. Stock market management
7. Parking automation
8. LibraryManagement
9. Online shopping
10. Content management

List of Course Seminar Topics:

1. Process Models
2. Requirement Engineering
3. Agile Methodology
4. Modelling using UML
5. Analysis and Design in OO systems
6. Principles and Practices of good Software Design
7. Collaborative software development
8. CMMI
9. Component diagram
10. Deployment diagram

Text Books: (As per IEEE format)

1. Hassan Gomaa, "Software Modeling and Design- UML, Use cases, Patterns and Software Architectures", Cambridge University Press, 2011, ISBN 978-0-521-76414-8
2. *Roger Pressman, "Software Engineering: A Practitioner's Approach"‖, McGraw Hill, ISBN 0–07–337597–7*

Reference Books: (As per IEEE format)

1. GardyBooch, James Rambaugh, Ivar Jacobson, "The unified modeling language user guide" , Pearson Education, Second edition, 2008, ISBN 0-321-24562
2. *2.* Ian Sommerville, "Software Engineering", Addison and Wesley, ISBN 0-13-703515-2

Course Outcomes:

The student will be able to –

1. Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices focusing tailored processes that best fit the technical and market demands of a modern software project.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual models using stakeholder analysis and requirement validation.
3. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework.
4. Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
5. Conform to Configuration Management principles and demonstrate cohesive teamwork skills avoiding classic mistakes and emphasizing on software safety adhering to relevant standards.
6. Analyze the target system properties and recommend solution alternatives by practicing project planning, scheduling, estimation and risk management activities.

Moocs Links and additional reading material:

www.nptelvideos.in

Future Courses Mapping:

1. Testing and Quality Assurance
2. Agile and DevOps Methodologies
3. User Interface and User Experience (UI/UX) Design
4. Security in Software Design

Job Mapping: (Optional)

Job opportunities that one can get after learning this course

1. Requirements Engineer
2. Software Architect
3. Software Designer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	1	3			3	2								
CO2	2	3		2	2			2			2		2	2
CO3	2	2		3	2									
CO4		3	3		3	2		2						
CO5	2	2							2					
CO6							2	2			2	3		
Average	2.25	2.6	3	2.5	2.5	2	2	2	2		2	3	2	2

FF No.: 654

MM0109A : Edge AI

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures : 2 Hrs/week	GD/PPT: 20 Marks
Practical : NA	HA: 10 Marks
Tutorial : 1 Hr/week	ESE:(W): 40 Marks

Course Prerequisites:

Artificial Intelligence, Machine Learning, Deep Learning

Course Objectives:

1. Understand fundamentals of Edge Computing and AI.
2. Explore edge architectures.
3. Understand Edge AI tools, virtualization, and DevOps.
4. Apply AI inference and training techniques at the edge.
5. Analyze Mobile Edge AI and TinyML deployment.
6. Design intelligent and secure Edge AI applications.

Course Relevance:

This course equips students with the knowledge and skills to develop, deploy, and optimize intelligent AI applications on resource-constrained edge devices for real-time and low-latency computing.

SECTION-I

Unit 1: Introduction to Edge Computing **(4 Hours)**

Definition and basic concepts of Cloud Computing; Evolution of computing: Mainframe →Client-Server → Cloud computing; Need for edge, Cloud-fog-edge comparison, AI at edge concept, Need for real-time AI, Edge AI Applications

Case Study: Smart Home Edge Computing System for real-time monitoring and automation.

Unit 2: Edge Computing Architectures **(5 Hours)**

Need for Edge Architecture; Core components: Edge devices (sensors, IoT devices), Edge nodes (gateways,

ıicro data centers), Cloud backend; Basic layered architecture: Device Layer, Edge Layer, Cloud Layer; Types of rchitectures (Device Edge Architecture, Fog Computing Architecture, Cloud-Edge Hybrid Architecture, Multi-ccess Edge Computing (MEC)) ; Resource constraints: Limited CPU, memory, power; Network considerations: onnectivity, bandwidth, Intermittent networks. Data management: Local processing vs cloud storage. Security & rivacy:Data protection at edge nodes.

Case Study : Edge Architecture for Smart Traffic Management.

Unit 3: Distributed Systems Concepts in Edge Computing (5 Hours)

Introduction to Distributed System; Why Distributed Systems are Essential in Edge Computing a Edge-specific hallenges: Dynamic nodes (IoT devices), Intermittent connectivity, Heterogeneity different devices); Cloud vs dge systems (comparison); Edge orchestration, Device coordination, d

distributed data sharing, fault tolerance in edge node

Case Study: Distributed Environmental Monitoring using IoT Edge Nodes.

SECTION-II

Unit 4: Edge AI Computing Tools (5 Hours)

Role of edge computing tools, Virtualization: Virtual Machine and Container, Docker, DevOps for Edge AI: CI/CD concepts, Pipeline workflow, Deployment automation

Case Study: Docker deployment on Jetson

Unit 5: Inference and Training in Edge AI (5 Hours)

Optimizing AI models in Edge: Quantization, Pruning, Understanding NVIDIA TensorRT format and its optimizations for inference, Federated Learning (FL) at Edge, Communication-Efficient FL

Case Study: Federated Learning for Privacy-Preserving Healthcare Applications.

Unit 6: Mobile Edge AI and Deployment of TinyML Model (4 Hours)

Deployment of TinyML models, Model conversion and deployment pipeline, Real-time inference on edge devices, Mobile AI, Resource-aware deployment

Case Study: TinyML-based Smart Agriculture using Arduino/ESP32.

List of Tutorials:

1. Study and comparison of Cloud Computing, Fog Computing, and Edge Computing with suitable use cases.
2. Design and explain the architecture of an Edge Computing system for a smart city or IoT application.
3. Analyze distributed system concepts such as logical clocks and clock synchronization using simple examples.
4. Install Git and GitLab and create a basic GitLab CI/CD pipeline using a sample. gitlab-ci.yml file.
5. Study virtualization technologies by comparing Virtual Machines (VMs) and Containers (Docker).
6. Explore Edge AI hardware platforms such as NVIDIA Jetson, Raspberry Pi, and Google Distributed Cloud Edge.
7. Convert and optimize a deep learning model using TensorFlow Lite (TFLite), ONNX, or TensorRT for edge inference.
8. Study Federated Learning architecture and compare it with centralized machine learning using a case study.
9. Deploy a TinyML model on an edge device (Arduino Nano 33 BLE Sense/Raspberry Pi Pico/ESP32) for real-time inference.
10. Present a case study on an Edge AI application such as Smart Healthcare, Smart Agriculture, Intelligent Transportation, or Industrial IoT.

List of Course Projects:

1. Smart Home Automation using Edge AI
2. Intelligent Traffic Monitoring using Edge Computing
3. Real-Time Face Mask/Face Detection on Raspberry Pi
4. Smart Agriculture Monitoring using TinyML
5. AI-Based Crop Irrigation System using Edge Devices
6. Diabetic Retinopathy Detection using Edge AI
7. Real-Time Object Detection using NVIDIA Jetson
8. Driver Drowsiness Detection using Edge AI

9. Smart Parking Management System using Edge Computing
10. Air Quality Monitoring using Edge AI
11. Human Activity Recognition using TinyML
12. Edge AI-Based Fire and Smoke Detection System
13. Industrial Equipment Fault Detection using Vibration Sensors
14. Smart Waste Management using IoT and Edge AI
15. Real-Time Helmet Detection for Traffic Safety
16. Intelligent Water Quality Monitoring System
17. Wildlife Monitoring using Edge AI Camera
18. Energy Consumption Prediction using Edge AI
19. Smart Healthcare Patient Monitoring using Edge Devices
20. Edge AI-Based Attendance System using Face Recognition

List of Home Assignments:

1. Compare Cloud Computing, Fog Computing, and Edge Computing with suitable real-world applications.
2. Explain the architecture of an Edge Computing system for a smart home, smart city, or healthcare application.
3. Study the challenges and advantages of distributed systems in Edge Computing and present a short report.
4. Compare Virtual Machines and Containers with respect to architecture, performance, and use cases.
5. Create a basic GitLab CI/CD pipeline using a sample .gitlab-ci.yml file and explain its workflow.
6. Study the hardware specifications and applications of Edge AI platforms such as NVIDIA Jetson, Raspberry Pi, and Google Distributed Cloud Edge.
7. Analyze TensorFlow Lite, ONNX, and TensorRT frameworks and compare their features for Edge AI deployment.
8. Explain the working of Federated Learning and discuss its advantages for privacy-preserving AI applications.

9. Study TinyML and describe the process of deploying a machine learning model on a resource-constrained edge device.
10. Prepare a case study on an Edge AI application in Smart Healthcare, Smart Agriculture, Intelligent Transportation, Smart Manufacturing, or Smart Cities.

List of Course Seminar Topics:

1. Edge Computing vs Cloud Computing
2. Edge AI in Smart Cities
3. Internet of Things (IoT) and Edge Computing
4. Virtualization using Docker and Containers
5. DevOps and CI/CD for Edge AI
6. NVIDIA Jetson and Raspberry Pi for Edge AI
7. TensorFlow Lite and TinyML
8. Federated Learning for Edge AI
9. Mobile Edge Computing (MEC)
10. Edge AI Applications in Healthcare and Smart Transportation

Text Books: (As per IEEE format)

1. D. Jadhav, P. Wani, N. Dasre, M. Niranjanamurthy, B. B. Mallik; 'Advanced Mathematics in Computing, Communication and Security'; 1st Edition; John Wiley & Sons Ltd; 2025.
2. M. Rovai; 'Edge AI Engineering: Hands-on with the Raspberry Pi'; —; Self-published (Open-source eBook); 2025.
3. Pete Warden, Daniel Situnayake, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers", O'Reilly Media.
4. Parikshit Narendra Mahalle, Mandar Pramod Diwakar, Vijaykumar Raghunath Ghule, Yashwant Sudhakar Ingle, "Edge: "Artificial Intelligence Algorithms, Applications, Challenges and Ethical Issues", Elsevier , 1st Edition.

Reference Books: (As per IEEE format)

1. D. Situnayake, J. Plunkett; 'AI at the Edge: Solving Real-World Problems with Embedded Machine Learning'; —; O'Reilly Media; 2023.
2. M. Khari, P. Gupta, C. P. Singh; 'Advancing Edge Artificial Intelligence: Transforming Computing from Cloud to Edge'; 1st Edition; CRC Press (Taylor & Francis Group); 2024.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press.

4. Daniel Situnayake, “AI at the Edge”, O’Reilly Media.

Course Outcomes:

The student will be able to –

1. Explain the concepts, evolution, and significance of Edge Computing and Edge AI.
2. Describe various Edge Computing architectures and distributed system concepts.
3. Demonstrate the use of virtualization and DevOps tools in Edge environments.
4. Apply AI inference optimization and distributed training techniques at the edge.
5. Analyze mobile Edge AI and TinyML deployment approaches under resource constraints.
6. Design efficient Edge AI solutions for real-world applications

Moocs Links and additional reading material:

1. MOOC Courses Links: R. Misra; ‘Edge Computing’; SWAYAM (NPTEL); https://onlinecourses.nptel.ac.in/noc26_cs30/preview; Accessed: Apr. 15, 2026
2. R. Misra; ‘Foundation of Cloud IoT Edge ML’; SWAYAM (NPTEL); https://onlinecourses.nptel.ac.in/noc23_cs65/preview; Accessed: Apr. 15, 2026
3. Introduction to Embedded Machine Learning (Coursera – Edge Impulse) https://www.coursera.org/learn/introduction-to-embedded-machine-learning
4. Fundamentals of TinyML (Harvard) https://pll.harvard.edu/course/fundamentals-tinyml
5. Edge AI Fundamentals (Edge Impulse) https://docs.edgeimpulse.com/knowledge/courses/edge-ai-fundamentals

Future Courses Mapping:

1. Advanced Artificial Intelligence
2. TinyML and Embedded AI
3. Cloud Computing and Distributed Systems
4. MLOps and Edge Analytics

Job Mapping:

1. Edge AI Engineer
2. AI/ML Engineer
3. IoT Engineer

4. Embedded AI Engineer
5. MLOps Engineer
6. Cloud & Edge Computing Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	1										1	3	
CO2	2	1	2										3	2
CO3	2	1	2		3							1	3	2
CO4	2	1	2	2	3								3	3
CO5	2	1	3	2	3							1	2	3
CO6	2	2	3	2	3			1	2	2		2	3	3
Average	1.83	2.33	2.00	1.00	2.00			0.17	0.33	0.33		0.83	2.83	2.17

FF No.: 654

MM0109: Natural Language Processing

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures : 2 Hrs/week	GD PPT: 20 Marks
Practical : NA	Home Assignment: 10 Marks
Tutorial : 1 Hrs/week	ESE Theory : 40 Marks

Course Prerequisites:

- Data structures
- Probability Theory and Statistics
- Basic knowledge of finite automata.

Course Objectives:

1. To introduce the fundamental concepts of Natural Language Processing (NLP), language modelling, and the different levels of language analysis.
2. To understand the linguistic foundations of NLP, including morphology, part-of-speech tagging, grammars, and parsing techniques.
3. To study semantic analysis techniques, lexical semantics, word sense disambiguation, and coreference resolution for natural language understanding.
4. To understand the principles and techniques used in major NLP applications such as information extraction, question answering, and text summarization.
5. To introduce the concepts, approaches, and evaluation techniques used in machine translation.
6. Promote critical thinking and interdisciplinary insight into how NLP is used across domains like sentiment analysis, chatbots, and multilingual processing.

Course Relevance:

This course equips students with the theoretical and practical foundations of Natural Language Processing by integrating concepts from computational linguistics, probability, machine learning, and formal language theory. Students learn classical NLP techniques including language modeling, sequence labeling, probabilistic parsing, semantic analysis, information retrieval, named entity recognition, question answering, summarization, and machine translation. The course develops the ability to design and evaluate systems that process unstructured textual data, supporting

applications such as search engines, intelligent assistants, information extraction, multilingual translation, and document summarization. Furthermore, the concepts learned in this course provide a strong conceptual foundation for understanding modern deep learning-based NLP and Large Language Model (LLM) architectures.

SECTION-I

UNIT 1 : Introduction To Natural Language Processing & Language Modelling (4 Hours)

The Study of Language, Applications of Natural Language Processing, Evaluation of Natural Language Processing Systems, The Different Levels of Language Analysis, Representations and Understanding, The Organization of Natural Language Processing Systems, Challenges in NLP , Introduction to language modelling, N-gram models, Word Embeddings: Introduction to Word2Vec and GloVe.

UNIT 2 : Sequential Tagging (4 Hours)

Morphology fundamentals, Finite State Machine Based Morphology, Introduction to POS Tagging, Hidden Markov Models for POS Tagging , Viterbi Decoding for Hidden Markov Models, Parameter Learning, Introduction to Maximum Entropy models and Conditional Random Fields.

UNIT 3 : Grammars And Parsing (5 Hours)

Grammars and Sentence Structure, What Makes a Good Grammar, Classical Parsing (Bottom-up, Top-down, Dynamic Programming: CYK parser) , Parsing using Probabilistic Context Free Grammars, Dependency grammar and parsing – introduction.

SECTION-II

UNIT 4 (5 Hours)

Fundamentals of Semantic Analysis, Meaning Representation, Syntax-driven Semantic Analysis, Lexical Semantics, WordNet. Word Sense Disambiguation (WSD): Introduction, Dictionary and Thesaurus Methods for WSD, Coreference Resolution: Anaphora, Cataphora, Reference Phenomena, Features for pronominal Anaphora Resolution, Pronominal Anaphora Baseline: The Hobbs Algorithm.

UNIT 5 (5 Hours)

Information Extraction: Named Entity Recognition, Information Retrieval (IR). Question Answering Systems: IR based Factoid Question Answering, Entity Linking, Knowledge Based Question Answering, Classic QA Models, Evaluation of Factoid Answers. Summarization: Summarizing single documents, Multi-Document Summarization, Content Selection in Multi-Document Summarization, Summarization Evaluation Techniques.

UNIT 6 **(5 Hours)**

Introduction to Machine Translation (MT), Classical MT & the Vauquois Triangle, Statistical MT, Phrase-Based Translation Model, Alignment in MT, MT Evaluation Techniques.

List of Assignments:

1. To learn to calculate bigrams from a given corpus and calculate probability of a sentence.
2. Using programming language Python and suitable libraries perform fundamental language processing for three different languages
3. Survey various techniques for POS tagging and implement any one of them
4. Write a program that can give you the country by its capital. Write a function that takes in three words, and the embeddings dictionary. Your task is to find the capital cities. For example, given the following words input: 1: Athens 2: Greece 3: Baghdad, your task is to predict the country 4: Iraq. You can use the pre-trained word embeddings and a similarity function.
5. Implement sentiment analysis over any suitable dataset by applying pre-processing, extracting necessary features and using machine learning algorithm

List of Tutorials:

1. Perform text preprocessing on a given text dataset using tokenization, stop-word removal, stemming, and lemmatization.
2. Apply regular expressions to clean and extract meaningful information from unstructured text data.
3. Implement Part-of-Speech (POS) tagging and chunking to analyze the grammatical structure of sentences using NLTK.
4. Develop a Named Entity Recognition (NER) system using spaCy to identify entities such as persons, locations, organizations, and dates.
5. Convert textual data into numerical feature vectors using Bag of Words (BoW) and TF-IDF techniques.
6. Study Sentence Segmentation and Syntactic Parsing using NLTK/spaCy
7. Generate word embeddings using the Word2Vec model and analyze semantic relationships between words.
8. Perform question answering using a pre-trained transformer model.

List of Course Projects:

1. **N-gram Language Model with Laplace Smoothing**
   Build bigram and trigram text predictors from scratch, implementing and evaluating basic absolute discounting and smoothing methods.
2. **Lesk Algorithm for Word Sense Disambiguation**

Create a program that determines the exact situational meaning of ambiguous words using dictionary overlap methods.

3. **Hobbs Algorithm for Pronominal Anaphora Resolution**
   Code the classic baseline Hobbs algorithm from scratch to resolve pronouns (like "he", "it") back to their correct nouns.
4. **Code-Mixed Sentiment Classifier**
   Classify Hinglish social media text into positive or negative sentiments using classical machine learning or lightweight Transformer models.
5. **Automated Resume Entity Parser**
   Extract key structural details like skills, education, and experience from unstructured resume text using custom spaCy NER models.
6. **Clickbait and Fake News Detector**
   Use a fine-tuned BERT-mini model to flag sensationalized or false news headlines via automated binary text classification.
7. **HMM Part-of-Speech Tagger**
   Build a fundamental statistical language tagger from scratch in pure Python using Hidden Markov Models and Viterbi decoding.
8. **English-to-Hindi Seq2Seq Translator**
   Develop a character or word-level translation model using sequence-to-sequence LSTMs trained on small parallel bilingual datasets.
9. **WordNet Similarity Engine**
   Compute semantic distance and lexical relationships between English words utilizing NLTK's built-in structural WordNet graph database interface.
10. **Local PDF Search (Mini RAG)**
    Query a local document and extract precise answers using text embeddings, ChromaDB vector store, and LangChain frameworks.
11. **E-Commerce Intent Classifier**
    Map user shopping queries to transactional actions by combining regex matching with cosine similarity over pretrained word embeddings.
12. **Extractive vs. Abstractive Summarizer**
    Compare traditional graph-based text compression algorithms against modern generative deep learning techniques using Hugging Face T5 models.
13. **LLM Prompt Evaluation Pipeline**
    Create a automated Python script testing multiple prompt variations across benchmark datasets to quantitatively measure LLM output accuracy.
14. **CYK Context-Free Grammar ParserSyllabus Match**
    Description: Implement the dynamic programming CYK algorithm in Python to parse sentences based on a custom Context-Free Grammar.
15. **Factoid Question Answering System**
    Build an Information Retrieval-based system that scans documents to isolate entity-linked answers for direct "Who/What/Where" user questions.

List of Course Seminar Topics:

1. **Smoothing Techniques in N-gram Modelling**
   Explores the mathematical adjustments used to prevent zero-probability errors when language models encounter unseen words during text evaluation.
2. **The Viterbi Algorithm in HMM POS Tagging**
   Explains the dynamic programming process used to decode and find the most accurate sequence of Part-of-Speech tags.
3. **MaxEnt Models vs. Conditional Random Fields (CRFs)**
   Compares these two sequence tagging methods, focusing on how they handle context features and avoid label-bias errors.
4. **The CYK Algorithm for Context-Free Parsing**
   Demonstrates how this dynamic programming algorithm builds structural sentence parse trees using grammars in Chomsky Normal Form.
5. **Constituency Parsing vs. Dependency Parsing**
   Contrasts traditional phrase-structure trees against modern syntax frameworks that map direct grammatical relationships between individual words.
6. **The Lesk Algorithm for Word Sense Disambiguation**
   Analyzes how dictionary definition overlaps are used to automatically determine the correct meaning of ambiguous contextual words.
7. **The Hobbs Algorithm for Anaphora Resolution**
   Reviews the classic, syntax-tree traversal rules used to trace and resolve pronouns back to their original nouns.
8. **Factoid vs. Knowledge-Based Question Answering**
   Compares text-scraping Information Retrieval systems against systems that extract direct factual answers from structured relational databases.
9. **Evaluation Challenges in Multi-Document Summarization**
   Examines redundancy filtering across multiple texts and details how automated metrics like ROUGE grade summary quality.
10. **The Vauquois Triangle in Machine Translation**
    Maps the architectural evolution of translation, from direct word substitution to deep syntactic and structural language transfers.

Text Books: (As per IEEE format)

1. *Daniel Jurafsky, James H. Martin, "Speech and Language Processing", Second Edition, Prentice Hall, 2008.*
2. *Christopher D.Manning and Hinrich Schutze, "Foundations of Statistical Natural Language Processing", MIT Press, 1999.*
3. *James Allen, "Natural Language Understanding", Pearson Publication, ISBN: 978-81-317-0895-8 2nd Edition*

Reference Books: (As per IEEE format)

1. Christopher D. Manning, HinrichSchutze, Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts, 1999
2. TanveerSiddiqui, US Tiwary, Natural Language Processing and Information Retrieval
3. Daniel M.Bikel, ImedZitouni, Multilingual Natural Language Processing Applications

Course Outcomes:

After completion of the course, student will be able to –

1. Explain the fundamental concepts, challenges, and applications of Natural Language Processing (NLP).
2. Analyze the morphological aspects of natural language and apply sequential tagging techniques for language processing.
3. Explain language grammars and apply parsing techniques for syntactic analysis of natural language.
4. Analyze semantic representations, lexical semantics, word sense disambiguation, and coreference resolution in natural language.
5. Apply NLP techniques to information extraction, question answering, and text summarization tasks.
6. Explain the principles of classical, statistical, and phrase-based machine translation, and evaluate machine translation systems using appropriate evaluation techniques.

Moocs Links and additional reading material: (If any)

- Natural Language Processing Specialization – DeepLearning.AI (Coursera) Instructor: Younes Bensouda Mourri, Łukasz Kaiser, and others https://www.coursera.org/specializations/natural-language-processing
- NPTEL: Natural Language Processing Instructor. Pawan Goyal|IIT Kharagpur https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_cs45

Future Courses Mapping:

1. Deep Learning for Natural Language Processing
2. Transformer-Based Language Models
3. Large Language Models (LLMs) and Prompt Engineering
4. Conversational AI and Intelligent Virtual Assistants
5. Knowledge Graphs and Semantic AI
6. Information Retrieval and Semantic Search
7. Machine Translation and Multilingual NLP

8. AI Agents and Language Technologies

Job Mapping:

1. NLP Engineer
2. AI Engineer
3. Machine Learning Engineer
4. Generative AI Engineer
5. Large Language Model (LLM) Engineer
6. Data Scientist (Text Analytics)
7. AI Research Engineer
8. Conversational AI Developer
9. Search & Information Retrieval Engineer
10. Software Engineer (AI/ML)
11. Knowledge Engineer
12. Applied AI Engineer
13. Language AI Engineer
14. Text Analytics Engineer
15. AI Solutions Developer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2			1							1	2	1
CO2	3	3			2							2	1	0
CO3	3	3	2	1	2							2	2	1
CO4	3	3	2	1	2							2	1	2
CO5	2	3	3	2	3				1	1		3	3	3
CO6	2	3	2	2	3							2	2	1
Average	2.7	2.8	2.3	1.5	2.2				1	1		2	1.8	1.6

FF No.: 654

AI 3206 : Design Thinking VI

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	0 Hrs/week	ESE: 100 Marks
Tutorial :	1 Hrs/week	

Course Prerequisites:

Basic knowledge of research work, research paper and patent.

Course Objectives:

1. Understand the concepts of design thinking approaches
2. Apply both critical thinking and design thinking in parallel to solve problems
3. Apply some design thinking concepts to their daily work
4. To provide ecosystem for students and faculty for paper publication and patent filing

Course Relevance:

The course is offered in S.Y. and T.Y. B.Tech. to all branches of Engineering.

Contents for Design Thinking :

Structure of The paper Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection tools Plagiarism

checking

Improving contents of the paper Patent drafting

Patent search Filing of patent

Writing answers to reviewer questions Modification in manuscript

Checking of publication draft

Assessment Scheme:

Publication of paper or patent

Course Outcomes:
On completion of the course, learner will be able

to–

CO1: Understand the importance of doing

Research

CO2: Interpret and distinguish different fundamental terms related to Research

CO3: Apply the methodology of doing research and mode of its publication

CO4: Write a Research Paper based on project work

CO5: Understand Intellectual property rights

CO6: Use the concepts of Ethics in Research

CO-PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	1	2	2	1	3	1	3	2	2	2	1	2	2
CO2	2	2	3	3	2	2	2	3	2	2	1	3	2	3
CO3	2	2	2	2	2	2	2	3	2	2	3	3	2	3
CO4	2	2	2	2	2	2	1	3	2	2	3	1	2	3
CO5	2	2	2	2	2	2	2	3	2	2	3	3	3	2
CO6	2	2	2	2	2	2	2	3	2	2	3	1	3	2
Average	2	1.83	2.16	2.16	1.83	2.16	1.66	3	2	2	2.5	2	2.33	2.5

FF No.: 654

AI3205 - Engineering Design & Innovation VI

Teaching Scheme		Examination Scheme
Credits:	2	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	4 Hrs/week	MSE : 30 Marks
Tutorial :	0 Hrs/week	ESE : 70 Marks

Course Prerequisites:

Problem Based Learning

Course Objectives:

7. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
8. To Evaluate alternative approaches, and justify the use of selected tools and methods,
9. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
10. To engage students in rich and authentic learning experiences.
11. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
12. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, the curriculum can be enriched with semester-long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defence, Education, Smart City, Smart Energy, and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy

SECTION I

Preamble - The content and process mentioned below is the guideline document for the faculti students to start with. It is not to limit the flexibility of faculty and students; rather they are fre their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory cou contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivit software technologies to solve real world problems using an interdisciplinary approach. The la form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology(Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intellige Human-Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc

Text Books: (As per IEEE format)

4. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
5. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
6. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro*

Reference Books: (As per IEEE format)

4. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.*
5. *Project management core textbook, second edition, Indian Edition , by Gopalan.*
6. *The Art of Agile Development. By James Shore & Shane Warden.*

Moocs Links and additional reading material:

www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from a societal need point of view

CO2: Choose and compare alternative approaches to select the most feasible one

CO3: Analyse and synthesize the identified problem from a technological perspective

CO4: Select the best possible solution to solve the problem.

CO5: Design & Develop a working model of the proposed solution.

CO6: Testing and validating product performance

Future Courses Mapping:

Major Project

Job Mapping:
Software Engineer. Software Developer, IT Engineer, Research Associate.

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	2			3				2	2			2
CO2				3	2	2		2					2	2
CO3	2		3		3			2	2				2	2
CO4		2				3							2	2
CO5			3				2		2				3	
CO6		2			3				2	3			2	
Average	0.83	1.16	1.33	0.5	1.33	1.33	0.33	0.66	0.5	0.83	0.33		1.83	1.33

B. Tech. Final Year Artificial Intelligence and Data Science

AY 2026-27 Module VII and VIII Course Content

FF No: 654

AI 4025 : High Performance Computing

Teaching Scheme		Examination Scheme
Credits:	2	CVV:30 Marks
Lectures :	2 Hrs/week	MSE: 30 Marks
Practical :	0 Hrs/week	HA : 10 Marks
Tutorial :	0 Hrs/week	ESE: 30 Marks

Course Prerequisites: Computer Organization, Operating System, Design & Analysis of Algorithms, Data Structure

Course Objectives:

Students will be able to

1. To introduce the basic concepts of High Performance Computing
2. To understand various GPU Architecture.
3. To write CUDA programs for parallel implementation
4. To organize the memory management in GPU
5. To optimize parallel programs on GPU using CUDA. To solve the scientific problems using GPUs

Course Relevance:

High Performance Computing, on the other hand, uses multiple processing elements simultaneously to solve a problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. This course is required in the industry & used to set up data centres.

SECTION-1

Introduction to Computing: High Performance, Parallel, Distributed; Motivation, Scope and Challenges; Parallelism vs Concurrency, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn's classification of multi-processors,, Amdhal's law; Parallel computer architectures : PRAM, Distributed memory systems, Shared memory systems and cache coherence, thread and process, Parallel computing architectures (multi-core CPUs, GPUs, traditional multi-processor system, Xeon-Phi, Jetson Kit, Kilocore processor), multiprocessor and multicomputer systems, interconnection networks, Modern GPU architecture, Performance comparison: Speedup, Gain time and scalability.

GPU architecture and parallel algorithms
Introduction to Modern GPU Tesla architecture, Types of GPU memories: global, shared, texture memory and their properties and uses, Streaming processor (SP), Streaming multiprocessor (SM), Special Functional unit (SFU), SM instruction types Fosters Parallel algorithm design, Designing GPU parallel algorithm for pattern clustering.

Programming Model: Common Unified Device Architecture (CUDA), CUDA programming model: threads, blocks, grid, Kernel, Kernel definition and kernel launch configuration, Use of GPU memories: global, shared, texture and constant memories, shared memory: organization, bank conflicts, global memory coalesced accesses, CUDA APIs: for memory allocation, synchronization, Execution of a CUDA kernel on GPU: concept of warp, warp divergence, CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc). Atomic operations in CUDA and their use.

SECTION-II

GPU Architecture: GPU architecture, Overview of the graphics pipeline, Components of GPU: Parallel streaming processors, Multiprocessors, Shared instruction caches ,Memory hierarchy – Global, Constant, Shared, and Texture memory; Case studies: NVIDIA Kepler K20/K40/K80/GP100/GV100/ Ampere.

Memory Organization and Optimization: Global, Shared, constant and texture memory. Memory coalescing, memory banks and bank conflicts, Page locked host memory. Reduction operation, CUDA code optimization. Need of profilers and analyzers, Introduction to CUDA Tools: MemCheck, Command line & Visual Profilers.

Scientific Computing and problem solving on GPU: Single vs. double precision, light weight scientific computing exercises, Image processing applications, Matrices etc. Parallel reduction on GPU and its applications. Compute intensive research-oriented problems and their GPU parallelization.

CUDA code optimization and Performance improvement CUDA code optimization: Memory optimization, Control flow optimization, Execution configuration optimization and Instruction optimization, Concept and application of page locked host memory, Single Vs. double precision computing on GPU: precision vss speed of computation, choosing correct precision for a real GPU application, memory leaks and associated problems, CUDA tools: cuda-memcheck and profiler.

List of Practical:

1. Parallel GPU implementation of vector-vector operations
2. Parallel GPU implementation of vector-Matrix operations
3. Parallel computation of binomial coefficient matrix
4. Parallel GPU implementation of Matrix-Matrix operations
5. Assignment focusing on optimization of data transfer between CPU and GPU: using page locked host memory and to avoid the data transfer
6. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.
7. Parallel GPU implementation involving kernel looping.
8. Use CUDA memcheck tool for knowing memory related errors in your source code
9. Profile your CUDA code using nvprof profiler tool for profiling your source code.
10. Write a program to know name of the GPU, its shared memory available and maximum CUDA block size.
11. Write a program to find the best GPU to execute your CUDA kernel, if multiple GPUs are connected to your system. Also set this device (GPU) for executing subsequent CUDA kernels.
12. A square matrix of size n x n contains either 1 or 0 in it. Write a CUDA kernel to compliment it without warp divergence.

List of project areas

The given list is indicative. A project area, other than listed here, can also be chosen but need to be mutually decided by student and teacher.

1. Pattern classification for large data sets
2. Clustering of patterns from large data set
3. processing of large images like MRI images
4. GPU Parallel acceleration of RDBMS queries using GPU
5. GPU Parallel acceleration of scientific tasks
6. GPU parallel acceleration of simulation of large systems
7. GPU parallel acceleration of global optimization algorithms
8. GPU parallel computations in Computer networks like cryptography, intrusion detection
9. GPU parallel computations in data analysis
10. Computationally intensive medical diagnosis
11. Regression analysis (linear and non-linear)
12. Artificial neural networks/deep learning/machine learning

List of Home Assignments:

Design:

1. Parallelizing Search Trees for Chess
2. Parallel Algorithm for Searching
3. Parallel Algorithm for sorting
4. Parallel Algorithm for Data mining
5.Parallel Algorithm for Image Processing

Case Study:

1. Nvidia DGX2
2. Jetson nano Developer Kit
3. GPU Accelerated Apache Spark
4. The Jetson Xavier NX Developer Kit
5. NVIDIA Ampere architecture

Blog

1. Cuda library
2. Turing mesh shaders
3. Low level GPU Virtual memory management
4. Memory Hierarchy of GPU
5. Comparison of Various GPUs

Surveys

1. Smart Hospitals through AI with GPUs
2. Clara Models to help fight with COVID 19
3. GPU Accelerated Molecular Dynamics Applications
4. Medical Imaging applications of GPU
5. Ray Tracing Applications of GPU

Suggest an assessment Scheme: MSE(30)+ESE(30)+HA(10)+CVV(20)

Text Books: (As per IEEE format)

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar; Introduction to parallel computing; second edition., Addison-Wesley, 2003, ISBN: 0201648652
2. David Kirk, Wen-mei HwuCUDA: Programming Massively Parallel Processors: A Hands-On Approach. © ELSEVIER Inc.
3. Jason Sanders and Edward KandrotCUDA by Example: An Introduction to General-Purpose GPU Programming"

Reference Books: (As per IEEE format)

1. Hwang and Briggs,"Computer Architecture and Parallel Processing", Tata McGraw Hill Publication ISBN 13: 9780070315563.
2. John Cheng, Max Grossman, Ty McKercherProfessional CUDA C Programming,
3. CUDA C PROGRAMMING GUIDE by NVIDIA

MOOCs Links and additional reading material:

www.nptelvideos.in
http://developer.nvidia.com/

Course Outcomes:

The student will be able to –

1) Recognize various parallel computing architectures and their fundamentals
2) Investigate parallel solutions to complex real world problems
3) Code the parallel programs on GPU using CUDA
4) Evaluate the performance on various GPU architectures

5) Optimize the parallel programs on GPU using CUDA
6) Design and develop new solutions to research problems

CO PO Map

CO1 –PO3(3) CO2 –PO5(3) CO3 –PO7(2) CO4 –PO11(1) CO5-PO12(1) CO6-PSO3(3)

CO attainment levels CO1 –3 CO2 -3 CO3 –2 CO4 –1 CO5-1 CO6-3

Future Courses Mapping: Parallel Computing, Distributed Computing

Job Mapping: What are the Job opportunities that one can get after learning this course Full Stack Architect-GPU Developer Technology Engineer Software Engineer Cloud Data Analytics Engineer Cloud Developer Senior Software Engineer HPC GPU Application Developer & Consultant GPU Programming Professional GPU Performance Analysis Lead / Architect GPU Advocate Associate

CO-PO Mapping :

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3									2			3	0
CO2		3				2	2	3						3
CO3	3		2		3				3					0
CO4				2										0
CO5	3						3					2		0
CO6		2	3	2		2	0				3			0
Average	1.5	0.833	0.833	0.66	0.5	0.66	0.833	0.5	0.5	0.33	0.5	0.33	0.5	0.5

FF No.: 654

AI4015: Network Security

Teaching Scheme		Examination Scheme
Credits:	2	CVV:30 Marks
Lectures :	2 Hrs/week	MSE: 30 Marks
Practical :	0 Hrs/week	HA : 10 Marks
Tutorial :	0 Hrs/week	ESE: 30 Marks

Course Prerequisites:

Computer Networks, Programming Fundamentals, Operating Systems, Basic Mathematics.

Course Objectives:

1. Understand the fundamental concepts of information and network security, including security goals, threats, vulnerabilities, attack types, and common software and protocol vulnerabilities.
2. Apply the principles of private-key cryptography. DES, and AES, Chinese Remainder Theorem,
3. Analyze public-key cryptographic techniques, and mathematical concepts, including modular arithmetic, prime numbers, Euclid's algorithm. RSA, Diffie–Hellman key exchange, and Elliptic Curve Cryptography (ECC), along with their security features and potential attacks.
4. Explain authentication mechanisms, hash functions, public key infrastructure, Kerberos, X.509 certificates, and operating system access control models for secure system design.
5. Understand network, transport, and application layer security protocols such as IPsec, SSL/TLS, HTTPS, S/MIME, PGP, honeypots, and security design principles for secure communication systems.
6. Develop an understanding of cybersecurity concepts, cyber-attacks, cybercrime trends, e-commerce security, and digital forensic techniques for investigating cyber incidents.

Course Relevance:

1. This course equips students with the knowledge and practical understanding required

to protect computer systems, networks, and digital information against modern cyber threats. It provides a strong foundation in cryptographic techniques, authentication mechanisms, secure communication protocols, access control models, and cybersecurity practices.

2. Students gain the ability to identify vulnerabilities, analyze security risks, and apply appropriate security solutions for real-world applications.
3. The course is highly relevant to current industry requirements, as cybersecurity has become a critical component of software development, cloud computing, IoT, e-commerce, banking, healthcare, and government systems. It also introduces students to digital forensics and cybercrime investigation, enabling them to understand incident response and forensic analysis.
4. Upon completion of the course, students will be prepared for roles such as:
    - Cybersecurity Analyst
    - Information Security Engineer
    - Network Security Engineer
    - Security Consultant
    - Penetration Tester (with additional specialized training)
    - SOC (Security Operations Center) Analyst
    - Digital Forensics Analyst
    - Security Auditor

SECTION-I

Unit 1: Introduction **(5 Hours)**

Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services.

key security properties - Confidentiality, Integrity, Availability.

Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, social engineering attacks, ransomware, SYN-Flooding, SQL- injection, DNS poisoning, Sniffing

Unit 2: Private key cryptography **(4 Hours)**

Transposition techniques: Rail Fence Transposition, Block (Single Columnar) Transposition, Double Columnar Transposition Cipher

Substitution techniques:

Monoalphabetic: Simple Substitution, Caeser Cipher.

Polyalphabetic: Vigenère Cipher, One-Time Pad (OTP)

Polygraphic Ciphers: Playfair Cipher, Hill Cipher.

Modern Block Ciphers: DES, AES. Chinese remainder theorem.

Unit 3: Public key cryptography **(5 Hours)**

Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), Role of random numbers in security, Importance of prime number.

RSA: RSA algorithm, Key generation in RSA, attacks on RSA.

Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.

Elliptic Curve Cryptography (ECC), Elliptic Curve arithmetic. Diffie-Hellman key exchange

SECTION-II

Unit 4: Authentication and access control **(5 Hours)**

Message authentication and Hash Function. Authentication: One-Way Authentication, Mutual Authentication, MD5, SHA-512, The Needham-Schroeder Protocol.

Kerberos, X.509 authentication service, public key infrastructure.

Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control.

Unit 5: **(5 Hours)**

Security application and design

Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack, Dictionary Attacks.

Network layer security: IPSec for IPV4 and IPV6.

Transport layer security: SSL and TLS.

Application layer security: Security services, S/MIME, PGP, Https, Honey pots.

Security design: End-to-end security, Security composability, Open design, Cost and tradeoffs

Unit 6: (4 Hours)

Cyber Security (Application and case studies)

Cyber Attack, Cyber Reconnaissance, Crimes in Cyber Space-Global Trends & classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digital forensic.

Home Assignments

1. Generate RSA keys using suitable prime numbers and demonstrate.
2. Explain applications of ECC in modern systems (e.g., HTTPS, blockchain, mobile security).
3. Design a secure network architecture for a given application by incorporating appropriate security protocols and access control mechanisms.
4. Study a recent cyber security incident and prepare a case study highlighting the attack methodology, investigation process, and preventive measures.
5. Evaluate authentication and access control techniques for different application domains and justify the most suitable approach.
6. Conduct a security assessment of a web application or networked system and recommend measures to improve its security posture.

Text Books: (As per IEEE format)

[1] W. Stallings, Cryptography and Network Security: Principles and Practice, 4th ed. Upper Saddle River, NJ, USA: Pearson Education, 2006.

[2] B. Menezes, Network Security and Cryptography, 1st ed. New Delhi, India: Cengage Learning, 2010.

[3] A. Kahate, Cryptography and Network Security, 4th ed. New Delhi, India: McGraw-Hill Education, 2019.

Reference Books: (As per IEEE format)

[1] M. Bishop, Computer Security: Art and Science, 1st ed. Boston, MA, USA: Addison-Wesley (Pearson Education), 2002. ISBN: 0201440997.

[2] C. Kaufman, R. Perlman, and M. Speciner, Network Security: Private Communication in a Public World, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002. ISBN: 9780130460196.

[3] V. K. Pachghare, Cryptography and Information Security, 2nd ed. New Delhi, India: PHI Learning, 2015. ISBN: 978-81-203-5082-3.

Course Outcomes:

The student will be able to –

1. Analyze cryptographic techniques using a mathematical approach by examining nature of attack.
2. Establish type of attack on a given system.
3. Identify different types of attacks.
4. Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
5. Design a secure system for protection from the various attacks for seven layer model by determining the need of security from various departments of an organization.
6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	3	3	0	0	0	0	2	0	0	0	0	0	0
CO2	2	2	3	0	2	0	0	0	0	0	0	0	0	3
CO3	3	2	0	0	2	0	0	0	2	2	0	0	0	2
CO4	0	0	2	2	0	0	2	0	0	0	0	3	0	2
CO5	3	2	0	0	0	0	2	0	0	0	0	0	2	0
CO6	0	0	0	3	0	2	0	2	0	0	2	0	0	0
Average	1.7	1.5	1.3	0.8	0.7	0.3	0.7	0.7	0.3	0.3	0.3	0.5	0.3	1.2

FF No. : 654

AI 4007: Reinforcement Learning

Teaching Scheme		Examination Scheme
Credits:	2	CVV:30 Marks
Lectures :	2 Hrs/week	MSE: 30 Marks
Practical :	0 Hrs/week	HA : 10 Marks
Tutorial :	0 Hrs/week	ESE: 30 Marks

Course Prerequisites: Proficiency in Python, Calculus, LinearAlgebra, Basic Probability and Statistics, Foundations of Machine Learning

Course Objectives:

1. To pursue basic knowledge of reinforcement learning techniques.
2. To understand foundation Techniques of Deep Reinforcement Learning.
3. To inculcate dynamic programming techniques.
4. To provide a clear and simple account of the key ideas and algorithms of reinforcement learning.
5. To explore how the learning is valuable to achieve goals in the real world.
6. To explore about how Reinforcement learning algorithms perform better and better in more ambiguous, real-life environments while choosing from an arbitrary number of possible actions.

Course Relevance: Reinforcement learning(RL) refers to a collection of machine learning techniques which solve sequential decision-making problems using a process of trial-and-error. It is a core are a of research in artificial intelligence and machine learning, and today provides one of the most powerful approaches to solving decision problems.

SECTION-1

Unit 1: The Reinforcement Learning Problem: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope (1 hour)

Unit 2: Finite Markov Decision Processes: The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, The Markov Property, Markov Decision Processes, Value Functions, Optimal Value Functions, Optimality and Approximation. (2 hours)

Unit 3: Dynamic Programming: Policy Evaluation, Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration, Efficiency

of Dynamic Programming. (3 hours)

Unit 4: Model-free solution techniques: Temporal difference learning, Monte Carlo Methods, Efficient Exploration and value updating. (2 hours)

SECTION-2

Unit 5: Batch Reinforcement Learning: Introduction, Batch Reinforcement Learning Problem, Foundations of Batch RL Algorithms, Batch RL Algorithms, Batch RL in Practice (2 hours)

Unit 6: Learning and Using Model: What is Model, Planning: Monte Carlo Methods, Combining Models and Planning, Sample Complexity, Factored Domains, Exploration, Continuous Domains, Empirical Comparisons, Scaling Up. (2 hours)

Unit 7: Planning and Learning with Tabular Methods: Models and Planning, Integrating Planning, Acting, and Learning, When the Model Is Wrong, Prioritized Sweeping, Full vs. Sample Backups, Trajectory Sampling, Heuristic Search, Monte Carlo Tree Search. (2 hours)

List of Course Seminar Topics:

1. Naive REINFORCE algorithm
2. 2.TD Control methods – SARSA
3. 3.Probability Primer
4. 4.Bellman Optimality
5. 5.Imitation learning
6. 6.Sequential Decision-Making
7. 7.Michael Littman: The Reward Hypothesis
8. 8.multi-agent learning
9. 9.An n-Armed Bandit Problem
10. 10.Q-Learning

List of Course Group Discussion Topics:

1. Human Intelligence versus machine intelligence
2. Security and Privacy in Pervasive Network
3. Security of Smart devices
4. Future of Ubiquitous Computing
5. Online Least-Square Policy Iteration

6. Gradient-Descent Methods
7. Bellman Optimality
8. Reward Shaping
9. Hierarchical RL
10. Atari Reinforcement Learning Agent

List of Home Assignments:

Design:

1. Smart personal health assistant

2. Human activities sensor

3. Intelligent buildings

4. Data storage searching in IOT

5. Protocols in IOT

Case Study:

1. Challenges in age of Ubiquitous computing

2. Ethnography in Ubiquitous computing

3. Cyber Physical System

4.Approaches to Determining Location Ubiquitous computing

5.Q-Learning for Autonomous Taxi Environment

Blog

1. Smart Devices for smart life
2. 2.Mobile affective computing
3. IOT and Cloud Computing
4. 4.Deep Q-Learning for Flappy Bird
5. 5.Q-Learning for any game Surveys
6. Data Collection for Ubiquitous computing Field
7. Usage of smart devices in daily lifestyle
8. Video Summarization
9. Behaviour Suite for Reinforcement Learning
10. 5. Causal Discovery with Reinforcement Learning

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360-degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE ESE PPT GD CVV HA

Text Books:(As per IEEE format)

1. Ed. John Krumm; Ubiquitous Computing Fundamentals; Chapman & Hall/CRC 2009

2.Richard S. Sutton and Andrew G. Barto, Reinforcement learning: An introduction, Second Edition, MIT Press, 2019

Reference Books:(As per IEEE format)

1. Wiering ,Marco, and Martijn Van Otterlo. Reinforcement learning. Adaptation, learning, and optimization 12 (2012)

2. Mohammad S. Obaidatandet al; Pervasive Computing and Networking, Wiley

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

The students should be able to

1) Define the key features of reinforcement learning that distinguishes it from AI and non-interactive machine learning
2) Formalize problems as Markov Decision Processes
3) Understand basic exploration methods and the exploration/exploitation trade-off
4) Understand value functions, as a general-purpose tool for optimal decision-making
5) Implement dynamic programming as an efficient solution approach to a real-world problem

Explain various tabular solution methods.

CO-PO Mappings:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3									2			3	0
CO2		3				2	2	3						3
CO3	3		2		3				3					0
CO4				2										0
CO5	3						3					2		0
CO6		2	3	2		2	0				3			0
Average	1.5	0.83	0.83	0.66	0.5	0.66	0.83	0.5	0.5	0.33	0.5	0.33	0.5	0.5

FF No.: 654

AI4028: Introduction to Large Language Models

Teaching Scheme	Examination Scheme
Credits: 2	Viva: 30 Marks
Lectures : 2 Hrs/week	MSE: 30 Marks
Practical : 0 Hrs/week	HA : 10 Marks
Tutorial : Hrs/week	ESE: 30 Marks

Course Prerequisites: Machine Learning, Deep Learning.

Course Objectives:

CO1: Understand the fundamental concepts of Natural Language Processing, statistical language models, and language representation techniques.

CO2: Apply deep learning techniques and neural architectures for solving Natural Language Processing tasks.

CO3: Analyze transformer architectures, pre-trained language models, and prompt engineering techniques for modern NLP applications.

CO4: Apply retrieval-augmented generation, knowledge graph techniques, and parameter-efficient adaptation methods for intelligent language systems.

CO5: Evaluate the capabilities, limitations, interpretability, and ethical implications of contemporary Large Language Models.

CO6: Develop practical NLP applications using PyTorch, Hugging Face, and modern transformer-based frameworks.

Course Relevance:

The basic ideas behind Large Language Models (LLMs) are presented in this course. It begins with an overview of the different NLP issues before going over how to use deep learning to solve the language modeling challenge. It explains the pre-training goals of the various Transformer-based models as well as the architectural nuances of Transformers. Recent developments in LLM research, such as LLM alignment, prompting, parameter-efficient adaptation, hallucination, prejudice, and ethical issues, are also covered. This course equips students to understand, evaluate, and tackle a range of LLM research issues.

SECTION-I

Unit 1: Introduction to Large Language Models (4 hours)
Introduction, Introduction to NLP (NLP Pipeline, Applications of NLP), Statistical Language Models, Statistical Language Models: Advanced Smoothing and Evaluation, Evaluation Metrics for Language Models.

Unit 2: Deep Learning for Natural Language Processing (4 Hours)

Introduction to Deep Learning, Perceptron, Artificial Neural Networks, Backpropagation Algorithm, Introduction to

PyTorch, Word Representation-Word2Vec, fastText, GloVe, Tokenization Strategies.

Unit 3: Large Language Models and Prompt Engineering (5 Hours)

Pre-training Strategies- ELMo, BERT (Encoder-only), Encoder–Decoder Models, Decoder-only Models. Introduction to Hugging Face, Instruction Tuning, Prompt-based Learning, Advanced Prompt Engineering Techniques, Prompt Sensitivity, Reinforcement Learning from Human Feedback (RLHF).

SECTION-II

Unit 4: Retrieval-Augmented Generation (5 hours)

Open-book Question Answering, Retrieval from Structured and Unstructured Knowledge Sources, Retrieval-Augmented Inference and Generation, Retrieval-Augmented Techniques - Key-Value Memory Networks, HotPotQA, Pointer Networks, Reading Comprehension, REALM, Retrieval-Augmented Generation (RAG), Fusion-in-Decoder (FiD), Unlimiformer.

Unit 5: Knowledge Graphs and Question Answering (5 hours)

Knowledge Graphs - Knowledge Graph Representation, Knowledge Graph Completion, Graph Alignment, Graph Isomorphism. Knowledge Graph Question Answering (KGQA) - EmbedKGQA, GrailQA. Graph Neural Networks vs Neural Knowledge Graph Inference.

Unit 6: Advanced Adaptation Techniques, Interpretability and Ethical NLP (5 hours)

Overview of Recent Large Language Models- OpenAI GPT-4, Meta Llama 3, Anthropic Claude 3, Mistral AI Mistral, Google Gemini. Ethical NLP - Bias and Fairness, Toxicity Detection, Responsible AI Practices, Limitations and Future Directions of Large Language Models.

List of Course Seminar Topics:

Evolution of Natural Language Processing

- Applications of NLP in Healthcare, Finance, and Education
- Statistical Language Models: N-grams and Smoothing Techniques
- Evaluation Metrics for Language Models (Perplexity, BLEU, ROUGE)
- Word Embeddings: Word2Vec vs. GloVe vs. fastText
- Tokenization Techniques in Modern NLP
- Deep Learning Architectures for NLP: ANN and CNN
- PyTorch Framework for NLP Model Development
- RNN, LSTM, and GRU: Comparative Study
- Sequence-to-Sequence Models and Neural Machine Translation
- Attention Mechanism and Multi-Head Attention
- Transformer Architecture: Design and Applications
- Pre-trained Language Models: ELMo, BERT, and T5
- Encoder-only, Decoder-only, and Encoder–Decoder Architectures
- Prompt Engineering Techniques for Large Language Models
- Reinforcement Learning from Human Feedback (RLHF)
- Retrieval-Augmented Generation (RAG): Architecture and Applications

- Knowledge Graphs for Intelligent Information Retrieval
- Question Answering Systems using Knowledge Graphs
- REALM, FiD, and Unlimiformer: Recent Advances in Retrieval-Augmented NLP
- Parameter-Efficient Fine-Tuning (LoRA, Prefix Tuning, Prompt Tuning)
- Interpretability and Explainability in Large Language Models
- Comparative Study of GPT-4, Llama 3, Claude, Gemini, and Mistral
- Ethical Challenges in NLP: Bias, Toxicity, Privacy, and Responsible AI

List of Course Group Discussion Topics:

Can Large Language Models Replace Traditional NLP Techniques?

- Transformers vs. Recurrent Neural Networks (RNN/LSTM): Which is Better for NLP?
- Is Prompt Engineering More Important than Model Fine-Tuning?
- Benefits and Challenges of Retrieval-Augmented Generation (RAG) in Real-World Applications
- Should Large Language Models Be Used in Education and Academic Assessment?
- Ethical Issues in Large Language Models: Bias, Fairness, and Toxicity
- Open-Source vs. Proprietary Large Language Models: Advantages and Limitations
- Role of Knowledge Graphs in Improving AI-Based Question Answering Systems
- Future of Multimodal AI: Integrating Text, Images, Audio, and Video
- Responsible AI: Balancing Innovation, Privacy, and Security in NLP Applications

List of Home Assignments:

1. **NLP Applications Review**
   Prepare a brief report on real-world applications of Natural Language Processing in different domains.
2. **Word Embeddings Comparison**
   Compare Word2Vec, GloVe, and fastText based on their working principles, advantages, and applications.
3. **Transformer Architecture**
   Draw and explain the Transformer architecture, including self-attention and positional encoding.
4. **Prompt Engineering Exercise**
   Design prompts for text summarization, translation, and question answering, and analyze the generated responses.
5. **Retrieval-Augmented Generation (RAG)**
   Study the RAG framework and explain its workflow, benefits, and applications.
6. **Ethical Issues in Large Language Models**
   Prepare a short report on bias, fairness, privacy, and responsible AI practices in NLP.

Text Books: (As per IEEE format)

1. T. Chakraborty, *Introduction to Large Language Models*, 1st ed. New Delhi, India: Wiley India, 2025. ISBN: 9789363864740.

2. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2008.

3. J. Eisenstein, *Natural Language Processing*, 1st ed. Cambridge, MA, USA: The MIT Press, 2019.

Reference Books: (As per IEEE format)

1. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (Online Draft).
2. Recent research papers from ACL, EMNLP, and NAACL conferences.
3. Recent research papers from NeurIPS, ICML, ICLR, AAAI, TACL, CL, JMLR, and TMLR.

Course Outcomes:

The student will be able to –

CO1: Explain the fundamental concepts of Natural Language Processing, statistical language models, and word representation techniques. **(L2 – Understand)**

CO2: Apply deep learning architectures and neural language models to solve Natural Language Processing problems. **(L3 – Apply)**

CO3: Analyze transformer architectures, attention mechanisms, and pre-trained language models for modern NLP applications. **(L4 – Analyze)**

CO4: Apply prompt engineering, retrieval-augmented generation (RAG), and knowledge graph techniques to develop intelligent language applications. **(L3 – Apply)**

CO5: Evaluate parameter-efficient fine-tuning methods, interpretability techniques, and the performance of contemporary Large Language Models. **(L5 – Evaluate)**

CO6: Develop NLP applications using PyTorch, Hugging Face, and transformer-based frameworks while considering ethical and responsible AI practices. **(L3 – Apply)**

Moocs Links and additional reading material:

1. https://www.coursera.org/specializations/natural-language-processing
2. https://www.coursera.org/learn/generative-ai-with-llms
3. Introduction to Large Language Models (LLMs) https://nptel.ac.in/courses/106102576

Future Courses Mapping:

Large Language Models and Generative AI, Knowledge Graphs and Semantic Web, Explainable and Responsible Artificial Intelligence, Machine Translation and Cross-Lingual NLP.

Job Mapping:

Natural Language Processing (NLP) Engineer, Machine Learning Engineer, Generative AI / Large Language Model

(LLM) Engineer, AI Research Engineer, Conversational AI (Chatbot & Virtual Assistant) Developer, Data Scientist (AI & NLP Applications).

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	-	-	2	-	-	-	-	-	-	2	2	1
CO2	3	3	2	2	3	-	-	-	1	-	-	2	3	3
CO3	3	3	2	3	3	-	-	-	-	-	-	2	3	3
CO4	2	3	3	2	3	1	-	1	1	1	-	2	3	3
CO5	2	3	2	3	2	2	1	3	-	1	-	2	2	2
CO6	2	2	3	2	3	2	1	3	2	2	2	3	3	3
	2.50	2.67	2.00	2.00	2.67	0.83	0.33	1.17	0.67	0.67	0.33	2.17	2.67	2.50

FF No.:654

AI4005: Major Project

Teaching Scheme		Examination Scheme
Credits:	9	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	20 Hrs/week	MSE: 30 Marks
Tutorial :	1 Hrs/week	ESE: 70Marks

Course Prerequisites:

Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Course Relevance:

Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course-based projects, the curriculum can be enriched with semester-long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy, and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to nalyze, design and apply categories of Bloom’s Taxonomy.

SECTION I

Preamble - The content and process mentioned below is the guideline document for the faculties an students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connecti of software technologies to solve real world problems using an interdisciplinary approach. The ladd the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology(Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence,Human-Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Securityetc).

Text Books: (As per IEEE format)

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
3. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert RobartCapraro, Mary Margaret Capraro*

Reference Books: (As per IEEE format)

1. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-basedlearning in engineering. Rotterdam: Sense Publishers. 2007.*
2. *Project management core textbook, second edition, Indian Edition , by Gopalan.*
3. *The Art of Agile Development. By James Shore & Shane Warden.*

Moocs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from a societal need point of view

CO2: Choose and compare alternative approaches to select the most feasible one

CO3: Analyse and synthesize the identified problem from a technological perspective

CO4: Select the best possible solution to solve the problem.

CO5: Design & Develop a working model of the proposed solution.

CO6: Testing and validating product performance

Job Mapping:

Software Engineer. Software Developer, IT Engineer, Research Associate.

CO - PO Mapping:

CO	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2		
CO1	3	3	2			3				2	2			2		
CO2				3	2	2		2					2	2		
CO3	2		3		3			2	2				2	2		
CO4		2				3							2	2		
CO5			3				2		2				3			
CO6		2			3				2	3			2			

FF No.: 654

AI 4023 : Design Thinking VII

Teaching Scheme		Examination Scheme
Credits:	1	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	0 Hrs/week	ESE: 100 Marks
Tutorial :	0 Hrs/week	

Course Prerequisites:

Basic knowledge of research work, research paper and patent.

Course Objectives:

1. Understand the concepts of design thinking approaches
2. Apply both critical thinking and design thinking in parallel to solve problems
3. Apply some design thinking concepts to their daily work
4. To provide ecosystem for students and faculty for paper publication and patent filing

Course Relevance:

The course is offered in S.Y. and T.Y. B.Tech. to all branches of Engineering.

Contents for Design Thinking :

Structure of The paper Journal List (Top 50 Journals)

Selection of the journal

Use of various online journal selection

tools Plagiarism checking

Improving contents of the paper

Patent drafting

Patent search Filing of patent

Writing answers to reviewer questions

Modification in manuscript

Checking of publication draft

Assessment Scheme:

Publication of paper or patent

Course Outcomes:

On completion of the course, learner will be able

to– CO1: Understand the importance of doing

Research

CO2: Interpret and distinguish different fundamental terms related to Research

CO3: Apply the methodology of doing research and mode of its publication

CO4: Write a Research Paper based on project work

CO5: Understand Intellectual property rights

CO6: Use the concepts of Ethics in Research

CO-PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	2	2	1	3	1	3	2	2	2	1	2	2
CO2	2	2	3	3	2	2	2	3	2	2	1	3	2	3
CO3	2	2	2	2	2	2	2	3	2	2	3	3	2	3
CO4	2	2	2	2	2	2	1	3	2	2	3	1	2	3
CO5	2	2	2	2	2	2	2	3	2	2	3	3	3	2
CO6	2	2	2	2	2	2	2	3	2	2	3	1	3	2
Average	2	1.83	2.16	2.16	1.83	2.16	1.66	3	2	2	2.5	2	2.33	2.5

FF No.:654

AI4008: Industry Internship

Teaching Scheme		Examination Scheme
Credits:	15	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	32 Hrs/week	MSE: 30 Marks
Tutorial :	0 Hrs/week	ESE: 70Marks

Course Relevance:

Implementation of technical knowledge acquired during previous three years of Internship and to get acquainted with Industry culture.

SECTION-I

Get used to corporate culture

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-II

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Neat project documentation

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE review for 100 marks converted to 70

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Explore career alternatives prior to graduation.

CO2: Integrate theory and practice.

CO3: Develop work habits and attitudes necessary for job success.

CO4: Develop communication, interpersonal and other critical skills in the job interview process.

CO5: Acquire employment contacts leading directly to a full-time job following graduation from college.

CO6: Practice Project Management and learn team dynamics

Moocs Links and additional reading material: (If any)

MOOCs:

- NPTEL/SWAYAM courses on Programming in C/C++, Java, Python, Data Structures, Algorithms, Operating Systems, Computer Networks, Database Management Systems, Software Engineering, Cloud Computing, Cybersecurity, and Web Technologies.
- Coursera, edX, Cisco Networking Academy, Microsoft Learn, AWS Skill Builder, Google Cloud Skills Boost, Oracle Academy, and Udemy courses aligned with Computer Engineering.

Coding Practice Platforms

- LeetCode
- HackerRank
- CodeChef
- GeeksforGeeks
- Codeforces

Documentation and Practice Resources

- Official documentation for C++, Java, Python, SQL, Git, Docker, Kubernetes, Linux, and Cloud platforms
- GitHub – Version control and open-source projects
- MDN Web Docs – HTML, CSS, JavaScript
- Oracle – Java Documentation
- Linux Foundation – Linux and DevOps learning resources

Job Mapping:

Software Engineer , Cloud Developer , Full Stack Developer , Security Analyst , Data Scientist , Business Analyst

AI Engineer , Data Scientist , Data Analyst , Machine Learning Engineer , Deep Learning Engineer , NLP Engineer

Generative AI Engineer , Data Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**
CO1	1	3	-	-	-	2	-	-	-	-	-	-	2	2
CO2	-	-	-	1	2	-	-	-	-	-	-	-	2	1
CO3	2	-	3	-	1	-	-	1	1	-	-	1	1	2
CO4	-	2	-	-	-	1	-	-	-	-	-	-	2	2
CO5	3	-	-	-	-	-	1	-	2	-	-	-	1	2
CO6	-	3	-	-	1	-	-	-	2	3	3	-	3	1
Average	1	1.33	0.5	0.16	0.66	0.5	0.16	0.16	0.833	0.5	0.5	0.16	1.83	1.66

FF No.:654

AI4011: International Internship

Teaching Scheme		Examination Scheme
Credits:	15	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	32 Hrs/week	MSE: 30 Marks
Tutorial :	0 Hrs/week	ESE: 70Marks

Course Relevance:

Implementation of technical knowledge acquired during previous three years of Internship and to inculcate research culture.

SECTION-I

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-II

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Research Paper should be published in Peer Reviewed Journal/Conference or Patent should be published.

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE review for 100 marks converted to 70

Course Outcomes:

The student will be able to –

CO1: Explore career alternatives prior to graduation.
CO2: Integrate theory and practice.
CO3: Develop work habits and attitudes necessary for job success.
CO4: Develop communication, interpersonal and other critical skills in the job interview process.
CO5: Acquire employment contacts leading directly to a full-time job following graduation from college. CO6: Practice Project Management and learn team dynamics

Moocs Links and additional reading material: (If any)

MOOCs:

- NPTEL/SWAYAM courses on Programming in C/C++, Java, Python, Data Structures, Algorithms, Operating Systems, Computer Networks, Database Management Systems, Software Engineering, Cloud Computing, Cybersecurity, and Web Technologies.
- Coursera, edX, Cisco Networking Academy, Microsoft Learn, AWS Skill Builder, Google Cloud Skills Boost, Oracle Academy, and Udemy courses aligned with Computer Engineering.

Coding Practice Platforms

- LeetCode
- HackerRank
- CodeChef
- GeeksforGeeks
- Codeforces

Documentation and Practice Resources

- Official documentation for C++, Java, Python, SQL, Git, Docker, Kubernetes, Linux, and Cloud platforms
- GitHub – Version control and open-source projects
- MDN Web Docs – HTML, CSS, JavaScript
- Oracle – Java Documentation
- Linux Foundation – Linux and DevOps learning resources

Job Mapping:

Software Engineer , Cloud Developer , Full Stack Developer , Security Analyst , Data Scientist , Business Analyst

AI Engineer , Data Scientist , Data Analyst , Machine Learning Engineer , Deep Learning Engineer , NLP Engineer Generative AI Engineer , Data Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO 2**
CO1	1	3	-	-	-	2	-	-	-	-	-	-	2	2
CO2	-	-	-	1	2	-	-	-	-	-	-	-	2	1
CO3	2	-	3	-	1	-	-	1	1	-	-	1	1	2
CO4	-	2	-	-	-	1	-	-	-	-	-	-	2	2
CO5	3	-	-	-	-	-	1	-	2	-	-	-	1	2
CO6	-	3	-	-	1	-	-	-	2	3	3	-	3	1
Average	1	1.33	0.5	0.16	0.66	0.5	0.16	0.16	0.833	0.5	0.5	0.16	1.83	1.66

FF No.:654

AI4010: Research Internship

Teaching Scheme		Examination Scheme
Credits:	15	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	32 Hrs/week	MSE: 30 Marks
Tutorial :	0 Hrs/week	ESE: 70Marks

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to inculcate Industry culture.

SECTION-I

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-II

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE review for 100 marks converted to 70

Course Outcomes: On completion of the course, learner will be able to–

CO1: Explore career alternatives prior to graduation.

CO2: Integrate theory and practice.

CO3: Develop work habits and attitudes necessary for job success.

CO4: Develop communication, interpersonal and other critical skills in the job interview process.

CO5: Acquire employment contacts leading directly to a full-time job following graduation from college.

CO6: Practice Project Management and learn team dynamics

Moocs Links and additional reading material: (If any)

MOOCs:

- NPTEL/SWAYAM courses on Programming in C/C++, Java, Python, Data Structures, Algorithms, Operating Systems, Computer Networks, Database Management Systems, Software Engineering, Cloud Computing, Cybersecurity, and Web Technologies.
- Coursera, edX, Cisco Networking Academy, Microsoft Learn, AWS Skill Builder, Google Cloud Skills Boost, Oracle Academy, and Udemy courses aligned with Computer Engineering.

Coding Practice Platforms

- LeetCode
- HackerRank
- CodeChef
- GeeksforGeeks
- Codeforces

Documentation and Practice Resources

- Official documentation for C++, Java, Python, SQL, Git, Docker, Kubernetes, Linux, and Cloud platforms
- GitHub – Version control and open-source projects
- MDN Web Docs – HTML, CSS, JavaScript
- Oracle – Java Documentation
- Linux Foundation – Linux and DevOps learning resources

Job Mapping:

Software Engineer , Cloud Developer , Full Stack Developer , Security Analyst , Data Scientist , Business Analyst

AI Engineer , Data Scientist , Data Analyst , Machine Learning Engineer ,Deep Learning Engineer , NLP Engineer

Generative AI Engineer , Data Engineer

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/ PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO12**	**PSO1**	**PSO2**
CO1	1	3	-	-	-	2	-	-	-	-	-	-	2	2
CO2	-	-	-	1	2	-	-	-	-	-	-	-	2	1
CO3	2	-	3	-	1	-	-	1	1	-	-	1	1	2
CO4	-	2	-	-	-	1	-	-	-	-	-	-	2	2
CO5	3	-	-	-	-	-	1	-	2	-	-	-	1	2
CO6	-	3	-	-	1	-	-	-	2	3	3	-	3	1
Average	1	1.33	0.5	0.16	0.66	0.5	0.16	0.16	0.8333	0.5	0.5	0.16	1.83	1.66

FF No.:654

AI4009: Project Internship

Teaching Scheme		Examination Scheme
Credits:	15	CP: NA
Lectures :	0 Hrs/week	CVV:NA
Practical :	32 Hrs/week	MSE: 30 Marks
Tutorial :	0 Hrs/week	ESE: 70Marks

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to get acquainted with Industry culture.

SECTION-1

Get used to corporate culture and get sponsorship from the company

Realization of Internship as per problem statement

Design, Testing / Experimentation, Analysis / Validation

Documentation and Report Writing

Quality of Work

Performance in Question & Answers Session

Regular interaction with guide

SECTION-2

Problem Statement

Literature Review

Clarity about the objectives of Internship activity

Requirement Analysis, Internship Planning

Knowledge of domain, Latest technology, and modern tools used /to be used

Neat project documentation

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE review for 100 marks converted to 70

Course Outcomes: On completion of the course, learner will be able to–
CO1: Explore career alternatives prior to graduation.

CO2: Integrate theory and practice.
CO3: Develop work habits and attitudes necessary for job success.
CO4: Develop communication, interpersonal and other critical skills in the job interview process.
CO5: Acquire employment contacts leading directly to a full-time job following graduation from college.
CO6: Practice Project Management and learn team dynamics

CO-PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1	3	-	-	-	2	-	-	-	-	-	-	2	2
CO2	-	-	-	1	2	-	-	-	-	-	-	-	2	1
CO3	2	-	3		1	-	-	1	1	-	-	1	1	2
CO4	-	2	-	-	-	1	-	-	-	-	-	-	2	2
CO5	3	-	-	-	-	-	1		2	-	-	-	1	2
CO6	-	3	-	-	1	-	-	-	2	3	3		3	1
Average	1	1.33	0.5	0.16	0.66	0.5	0.16	0.16	0.833	0.5	0.5	0.16	1.83	1.66