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 2025-26

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)

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.

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 25-26)
Index

Sr.No	Course Code	Title/Course Name	Page No.
1	-	Institute Vision, Mission	-
2	-	Department Vision , Mission	-
3	-	Program Educational Objectives (PEO)	--
4	-	Programme Outcomes (PO)	-
5	-	Programme Specific Outcomes (PSO)	-
6	-	Nomenclature for Teaching and Examination Assessment Scheme AY 2024-25	-
7	-	Structure - Module III	-
8	-	Structure -Module IV	-
9	-	Structure -Module V	-
10	-	Structure -Module VI	-
11	-	Structure -Module VII (Department Module , Internship)	-
12	-	Structure - Module VIII (Department Module , Internship)	-
13	AI2301	Data Structures	2
14	AI2302	Foundations of Intelligent Digital System	6
15	AI2303	Database Management System	14

16	AI2304	Discrete Structures and Computational Theory	**21**
17	AIM001/AIM002/ AIM003	MDM(ARVR, IP, CG)	**25**
18	HS2002	From Campus to Corporate	**37**
19	HS2001	Reasoning and Aptitude Development-3	**39**
20	AI2305	Design Thinking- I	**42**
21	AI2306	Engineering Design & Innovation – I	**44**
22	AI2307	Object Oriented Programming	**47**
23	AI2308	Computer Network	**56**
24	AI2309	Mathematical foundations for Artificial Intelligence	**62**
25	AI2310	Probability and Applied Statistics	**66**
26	MM0102/MM0103/MM0104	MDM (ARVR, IP, CG)	**69**
27	AI2311	Design Thinking II	**79**
28	AI2312	Engineering Design & Innovation II	**81**
29	AI3001	Artificial Intelligence	**85**
30	AI3002	Operating System	**90**
31	AI3003	Statistical Inference	**97**
32	AI3004	Machine Learning	**104**
33	AI 3005	Engineering Design & Innovation – V	**111**
34	AI 3013	Design Thinking V	**114**
35	-	Coursera*	**-**
36	AI3011	Complexity Algorithms	**116**

37	AI3012	Software Design and Methodologies	**123**
38	AI3010	Deep learning	**130**
39	AI 3008	Engineering Design & Innovation –VI	**135**
40	AI3015	Design Thinking VI	**138**
41	AI4025	High Performance Computing	**141**
42	AI4015	OE2: Network Security	**145**
43	AI4007	OE 3: Reinforcement Learning	**148**
44	AI4005	Major Project	**152**
45	AI4023	Design Thinking VII	**155**
46	AI4008	Industry Internship	**157**
47	AI4011	International Internship	**159**
48	AI4010	Research Internship	**161**
49	AI4009	Project Internship	**163**

Nomenclature for Teaching and Examination Assessment Scheme AY 2025-26

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: 2025-26 Module: III&IV

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											Credits
			Theory	Lab	Tut	In Semester Assessment				End Semester Assessment						Total	
						Lab CA	T1	MSE	T2	ESE Pract	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI2301	Data Structures	2	4	0	10	-	-	-	40	-	-	20	-	30	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	0	0	-	35(W)	-	35(W)	-	-	-	30	-	-	100	2
S5	AIM001/ AIM002/ AIM003	MDM(ARVR, IP, CG)	2	0	1	-	35(O)	-	35(O)	-	-	-	30	-	-	100	3
S6	HS2002	From Campus to Corporate	2	0	0	-	-	50(O)	-	-	-	50(O)	-	-	-	100	2
S7	HS2001	Reasoning and Aptitude Development-3	-	-	-	-	-	-	-	-	-	-	-	-	-	100	1
S8	AI2305	Design Thinking- I	-	-	1	-	-	-	-	-	-	-	-	-	-	100	1
S9	AI2306	Engineering Design & Innovation – I	-	4	-	-	-	30	-	-	-	70	-	-	-	100	2
Total			12	12	2	30	70	80	70	40	80	120	120	-	120	900	21

Title: Course Structure **SEM-II** **FF No.653**
Branch: AI&DS Year: SY AY: 2025-26 Module: III&IV

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Assessment Scheme (100 mark scale)											Credits
			Theory	Lab	Tut	In Semester Assessment				End Semester Assessment						Total	
						Lab CA	T1	MSE	T2	ESE Pract	ESE (W)	ESE (O)	CVV	GD/ PPT/ HA	CP		
S1	AI2307	Object Oriented Programming	2	4	0	10	-	-	-	40	-	-	20	-	30	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	-	20	-	30	100	3
S4	AI2310	Probability and Applied Statistics	2	0	0	-	35(W)	-	35(W)	-	-	-	30	-	-	100	2
S5	MM0102/ MM0103/ MM0104	MDM(ARVR, IP, CG)	2	0	1	-	35(O)	-	35(O)	-	-	-	-	-	30	100	3
S6	HS2003	From Campus to Corporate	2	0	0	-	-	50(O)	-	-	-	50(O)	-	-	-	100	2
S7	HS2004	Reasoning and Aptitude Development -3	-	-	-	-	-	-	-	-	-	-	-	-	-	100	1
S8	AI2311	Design Thinking- II	-	-	1	-	-	-	-	-	-	-	-	-	-	100	1
S9	AI2312	Engineering Design & Innovation – II	-	4	-	-	-	30	-	-	-	70	-	-	-	100	2
Total			12	12	2	30	70	80	70	40	80	120	120	-	120	900	21

Title: Course Structure **SEM-I** **FF No.653**
Branch: AI&DS Year: TY AY: 2025-26 Module:V

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)		Examination scheme									
						CA	MSA	ESA						
			Theory	Lab	Tut	Lab	MSE	CP	HA	GD/ PPT	ESE	CV V	Total	Credits
S1	AI3001	Artificial Intelligence	2	2	1	10	-	20	-	20	30	20	100	4
S2	AI3002	Operating System	2	2	1	10	-	20	-	-	50 (Lab)	20	100	4
S3	AI3003	Statistical Inference	2	2	1	10	-	20	20	-	30	20	100	4
S4	AI3004	Machine Learning	2	2	1	10	-	20	-	20	30	20	100	4
S5	AI 3005	Engineering Design & Innovation – V	-	12	-	-	30	-	-	-	70	-	100	6
S6	AI3013	Design Thinking- V	-	-	1	-	-	-	-	-	100	-	100	1
S7	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	100	-	100	1
Total			8	20	5	40	30	80	20	40	410	80	700	24

Audit Courses for Third Year:

1. Mainframe Technologies
2. Basics of Game Development

Title: Course Structure **SEM-I** **FF No.653**
Branch: AI&DS Year: TY AY: 2025-26 Module:VI

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme								
						CA	MSA	ESA						
			Theory	Lab	Tut	Lab	MSE	CP	HA	GD/ PPT	ESE	CVV	Total	Credits
S1	AI3002	Operating System	2	2	1	10	-	20	-	-	50 (Lab)	20	100	4
S2	AI3004	Machine Learning	2	2	1	10	-	20	-	20	30	20	100	4
S3	AI3011	Complexity Algorithm	2	2	1	10	-	20	20	-	30	20	100	4
S4	AI3012	Software Design and Methodologies	2	2	1	10	-	20	-	20	30	20	100	4
S5	AI3008	Engineering Design & Innovation –VI	-	12	-	-	30	-	-	-	70	-	100	6
S6	AI3015	Design Thinking-VI	-	-	1	-	-	-	-	-	100	-	100	1
S7	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	100	-	100	1
Total			8	20	5	40	30	80	20	40	410	80	700	24

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

Branch: AI&DS Year: TY AY: 2025-26 Module: V

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)		Examination scheme								Total	Credits
						CA	MSA	ESA						
			Theory	Lab	Tut	Lab	MSE	CP	HA	GD/ PPT	ESE	CVV		
S1	Coursera*	Coursera*	2	-	-	-	-	-	-	-	-	-	100	4
S2	AI3001	Artificial Intelligence	2	2	1	10	-	20	-	20	30	20	100	4
S3	AI3003	Statistical Inference	2	2	1	10	-	20	20	-	30	20	100	4
S4	AI3010	Deep Learning	2	2	1	10	-	20	-	20	30	20	100	4
S5	AI3013	Engineering Design & Innovation – V	-	12	-	-	30	-	-	-	70	-	100	6
S6	AI3005	Design Thinking- V	-	-	1	-	-	-	-	-	100	-	100	1
S7	SH2001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	100	-	100	1
	Total		8	18	4	40	30	80	20	40	360	80	700	24

Title: Course Structure **SEM-II** **FF No.653**
Branch: AI&DS Year: TY AY: 2025-26 Module: VI

Sr. No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination scheme								Credits
						CA	MSA	ESA					Total	
			Theory	Lab	Tut	Lab	MSE	CP	HA	GD/ PPT	ESE	CVV		
S1	Coursera*	Coursera*	2	-	-	-	-	-	-	-	-	-	100	4
S2	AI3010	Deep Learning	2	2	1	10	-	20	-	20	30	20	100	4
S3	AI3011	Complexity Algorithm	2	2	1	10	-	20	20	-	30	20	100	4
S4	AI3012	Software Design and Methodologies	2	2	1	10	-	20	-	20	30	20	100	4
S5	AI3008	Engineering Design & Innovation – VI	-	12	-	-	30	-	-	-	70	-	100	6
S6	AI3015	Design Thinking- VI	-	-	1	-	-	-	-	-	100	-	100	1
S7	SH2001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	100	-	100	1
Total			8	18	4	30	30	60	20	40	360	60	700	24

Coursera Course List:

Sr No	Course Code	Name
1	MD3101	IBM Full Stack Software Developer
2	MD3103	IBM Back-End Developer
3	MD3106	Microsoft Power BI Data Analyst
4	MD3110	IBM Cybersecurity Analyst
5	MD3112	Google Cybersecurity
6	MD3115	IBM Data Analytics with Excel and R
7	MD3116	IBM Data Engineering
8	MD3117	Meta Database Engineer
9	MD3120	IBM Data Warehouse Engineer
10	MD3121	IBM DevOps and Software Engineering
11	MD3124	IBM Front-End Developer
12	MD3126	Meta iOS/Android Developer
13	MD3130	IBM Mainframe Developer
14	MD3132	Akamai Network Engineering
15	MD3140	SAP Technology Consultant
16	MD3141	AWS Cloud Technology Consultant
17	MD3142	Google UX Design
18	MD3148	Tableau Business Intelligence Analyst

Title: Course Structure **FF No.653**
Branch: AI&DS **Year: B. Tech AY: 2025-26**
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	AI4011	International Internship	-	32	-	-	-	30	-	70	-	-	100	15
S3	AI4010	Research Internship	-	32	-	-	-	30	-	70	-	-	100	15
	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 2025-26

FF No.: 654

AI2301: Data Structures

Credits: 04 **Teaching Scheme Theory: 02Hours / Week**

Lab: 04 Hours/ Week

SECTION-I

Section 1: Arrays, Stack, Queue, Linked List

Single and Multidimensional arrays, Time & Space Complexity Analysis.

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

Search techniques: Binary search, Fibonacci search.

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 Schedulingetc.

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

SECTION-II

Section2: Trees, Graphs, Hashing

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)

Graph: -Terminology and representation, Traversals, connected components and Spanning trees: Primes and Kruskal's Algorithm, Shortest Paths and Transitive Closures: Single Source All

destinations (Dijkstra's Algorithm), All pair shortest path algorithm, Topological Sort. **Hashing: -** Hashing techniques: Hash table, Hash functions, and Collision, Cuckoo Hashing.

List of Practical's: (Any Six)

1. Assignment based on Sorting and Searching.
2. Assignment based on Stack Application
3. Assignment based on Queue Application
4. Assignment based on different operations on Singly, Doubly, Circular linked list.
5. Assignment based on BST operations
6. Assignment Based on TBT operations
7. Assignment based on AVL tree
8. Assignment based on DFS and BFS
9. Assignment based on Prim's.
10. Assignment Based on Kruskals.
11. Assignment based on Shortest path problem
12. Assignment based on Hashing.
13. Assignments based on min max Heap tree.
14. AI-based assignment to generate test cases for data structure implementations
15. AI-based assignment to generate visualization of advanced trees.
16. AI-based assignment to implement decision tree for various applications.

List of Course Projects:

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 engines using 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 understand the properties of data structures with their memory representations.
2. To use linear data structures like stacks, queues etc. with their applications .
3. To implement linked lists for dynamic memory management.
4. To analyze tree data structures for hierarchical data representation and operations.
5. To understand Graph data structure with representation and traversal algorithms.
6. To justify the appropriate data structure in real life 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	
CO2		3			2							2		
CO3	3		3	2										
CO4	2	2	3	2	2		2		2					
CO5		3	3	3	2				2		2			
CO6			3			3		2		2				3
Average	1.33	1.6	2	1.16	1.33	0.5	0.3	0.3	0.6	0.3	0.3	0.3	0.3	0.5

FF No 654

AI 2302: Foundations of Intelligent Digital Systems

Course Prerequisites:

1. Basic electronics system

Course Objectives:

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

Credits: 03 **Teaching Scheme** **Theory:**02Hours/Week
Lab: 02 Hours/Week

Course Relevance:

The course is offered in S. Y. B.Tech (AI & DS) Engineering

Digital Electronics and Microprocessor Subject is aimed at developing a deep understanding of digital electronic circuits. At the end of the course, one would be able to analyze and synthesize different kind of combinatorial and sequential digital systems for real-world use.

This course also

consists of Microprocessors 8086, 8086 microprocessors are used in many other electronic devices, including cell phones, kitchen appliances, automobile emission-control and timing devices, electronic games, telephone switching systems, thermal controls in the home, and security systems.

SECTION-I

Digital Fundamentals

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, Karnaugh map Minimization

(4Hours)

Combinational Digital Circuits:

Standard representation for logic functions, simplification of logic functions using K-map, minimization of logical functions. Dont care conditions.BCD code,Gray code, Excess-3 code, Code converter, 4 bit binary adder & Sub-tractor:7483, Multiplexers & De-multiplexers, Encoder: Priority encoders, Decoders: 74138,ALU: 74181,Parity generator and checker. **(5Hours)**

Sequential Circuit:

Introduction of flip-flop (F.F), 1 bit memory cell, clocked S-R, J-K, T, DFlip-flop: Truth table, Excitation table, Characteristics table, Shift Register, Asynchronous and Synchronous counter, Sequence Generator, Sequences detector (Moore and Mealy). **(5Hours)**

SECTION-II

Introduction to 8086 microprocessor:

Internal Architecture, Generation of physical address8086,8086 memory segmentation, Register Organization, Addressing modes: Immediate addressing, Register addressing,

Direct addressing, Indirect addressing, Relative addressing, Indexed addressing, Bit inherent addressing, bit direct addressing.

(4Hours)

8086 Instructions types

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,

(5Hours)

Interrupt Structure and Programmable Interval Timer:

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 Interval timer/counter) block diagram, control word, Operating modes of 8259, Interfacing with 8086 processor and Programming.

(5 Hours)

List of Tutorials:

1. AI-Optimized Memory Systems for Intelligent Computing
2. GPU Computing for AI: CUDA in Deep Learning and Neural Networks
3. AI-Based Cache and Memory Replacement Algorithms
4. AI-Driven Instruction Pipelining for Performance Optimization
5. RISC Vs CISC architecture: A Case Study
6. ARM processor architecture
7. Latest Technology in Embedded systems

8. Virtual Memory Management Enhanced with AI Techniques

9. Booth's Encoding Pattern for Fast Scalar Point Multiplication in ECC for Wireless Sensor Networks

10.Machine Learning Approaches to Cache Coherence in Multicore Systems

11. State of the art parallel processor design.

12. Memory management in mobile OS.

13. Evolution of processors.

List of Practical's: (Any Six)

1. Simulation and Verification of Logical Gates and Boolean Algebra using AI Tools.

2. AI-assisted Code Converters (Excess-3 to BCD and vice versa).

3. Implement 16:1 MUX using 4:1 MUX (IC 74153), and use AI (e.g., Genetic Algorithms) to optimize the selection logic for speed or gate minimization.

4. Intelligent Decoder System – 2-bit Comparator Using IC 74138.

5. Synchronous Up/Down Counter using JK Flip-Flop with AI Simulation.

6. AI-driven Sequence Detector using JK Flip-Flop

7. Study of 8086 Architecture with AI-based Visualization Tools.

8. 8086 ALP: Count Positive/Negative Numbers and Use AI for Sentiment Prediction Analogy.

9. 8086 ALP: Marks Analysis with AI-based Grading System.

10. 8086 ALP for Block Transfer – Compared with AI Data Preprocessing Techniques.

11. 8086 ALP for String Operations a. find length. b. Reverse of string c. Palindrome

List of Course Projects:

1. AI-Powered Weather Imaging CubeSat with Smart Telemetry Transmission

2. Smart Ebike Speed Controller System using AI & IoT

3 AI-Based Air and Water Pollution Sensing Smart Watch.

4. Solar-Powered Sea Weather & Pollution Transmitter Buoy with AI Analytics

5. AI-Enabled Coin Operated Water ATM with Smart Bottle Dispenser

6. Multiple Cities Load Shedding Using ARM

7. AI-Driven Wireless Biomedical Parameter Monitoring System (ARM9)

8. Smart Home/Office/Industrial Security System using AI, ARM & RFID

9. AI-Augmented Electronic Voting Machine (EVM) using ARM

10. AI-Based Online Parallel Examination System

List of Course Seminar Topics:

1. GPU computing: CUDA

2. AI-Oriented Parallel Computing Models and Frameworks

3. From Traditional to AI-Powered Modern Computer Generations

4. AI in Supercomputing: Accelerating Scientific Discovery

5. Virtual Memory Management Enhanced with AI Techniques

6. Replacement Algorithms

7. Pipelining

8. Cache Coherence

9. Super Computer

10. Intelligent Hazard Detection and Resolution in Pipelining Using AI

Text Books: (As per IEEE format)

1. G.K Kharate 'Digital Electronics' – Oxford University Press 2010 seventh impression 2013

2. Douglas Hall, "Microprocessors and Interfacing",2nd Edition, Tata McGraw Hill Publications, ISBN 0-07-025742-6.

3. "Advanced 80386, programming techniques " , James Turley , Tata McGraw Hill Publications, ISBN –

0-07-881342-5

4. Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630- 011, December 1995. 5. R.P. Jain, "Modern Digital Electronics,"3 rdEdition, 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:

1. Upon completion of the course, student will be able to – 1. Learn and illustrate the standard representation for logical functions
2. Explore the knowledge of Digital logic circuits.
3. Design applications based on combinational and sequential circuits.
4. Understand and analyze the architecture of digital systems and evaluate their relevance in designing AI accelerators like TPUs, GPUs, and FPGA-based systems.
5. Adapt the knowledge based on microprocessor instructions
6. Illustrate the concept of interrupts and its service routine

Future Courses Mapping:

1. Deep Learning

2. Reinforcement Learning

3. DBMS

4. Big Data

5. Data Mining

6. Information Retrieval

7. Recommendation Systems

8. Cloud Computing – AWS

9. IOT

10. Artificial Intelligence

11. Pattern Recognition

12. Natural Language Processing

13. Computer Vision

Job Mapping:

Job opportunities that one can get after learning this course

1. Data Scientist

2. Data Analyst

3. AI Engineer

4. Data Architect.

5. Data Engineer.

6. Statistician.

7. Database Administrator.

8. Business Analyst

9. Business Intelligence Developer

10. Infrastructure Architect

11. Enterprise Architect

12. Machine Learning Engineering

13. Machine Learning Scientist

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**	**PSO1**	**PSO2**
CO1	2	1	2		1			1			1	1	1	1
CO2	2	2	2		1	2					1	1	1	1
CO3	2		2		1	2					1	1		1
CO4	2		2		1	2		1			1	1		1
CO5	2	1	2			2		1			1	1	1	
CO6	2										1	1	1	
Average	2	0.6	1.66		0.6	1.33		0.5			1	1	0.6	1.6

FF No: 654

AI 2303: DATABASE MANAGEMENT SYSTEMS

Course Prerequisites:

1 Data structures

2. Discrete Mathematics

Course Objectives:

1. Understand core concepts of database systems, including data models, ER modelling, relational model, and schema design principles with normalization up to 4NF.
2. Develop efficient SQL queries using basic and advanced constructs such as subqueries, joins, views, procedures, functions, and triggers.
3. Analyse and optimize query performance using relational algebra, indexing methods, and query execution plans.
4. Apply principles of transaction management and concurrency control, ensuring database consistency through ACID properties, serializability, and recovery techniques.
5. Explore storage structures and file organization techniques, including B+ Trees and hashing, sand understand their impact on query processing.
6. Gain exposure to modern database technologies, including NoSQL, cloud, mobile databases, and hands-on experience with MongoDB and SQLite

Credits: 03 **Teaching Scheme Theory:** 02Hours/Week

Lab: 02Hours/Week

Course Relevance:

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

SECTION-I

Topic and Content (01 Hours)

Introduction to Database Systems: Need of DBMS, Design of DBMS, characteristics, components of DBMS, DBMS vs File Systems, Applications of DBMS in AI/ML

Data Models and Relational Model: Types of data models ,Relational model: schema, keys, integrity constraints.

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

Relational Algebra & SQL Basics:Set, string, Date and numerical function, selection, projection, SQL: DDL, DML, DQL, Aggregate Function, Group by and having clause.

Advanced SQL: Subqueries, Procedure, Function, Trigger, Cursors, joins, views, Nested Queries.

SECTION-II

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

Transaction Management & Concurrency Control: Basic Concept, ACID properties, State diagram, Concept of Schedule, Serial Shedule, Serializability-Conflict and View, Concurrency control Protocols, Recovery techniques: Shadow Paging, Log based recovery, check Point.

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

Emerging database Technologies: Internet Databases, Mobile databases, Cloud Databases, SQLite Database, I/O Parallelism, Distributed Database Systems.

List of Course Seminar Topics:

1. Object and Object-Relational Databases
2. XML data model, XML documents and associated languages
3. Database Security
4. Modern Storage Architectures
5. Google Cloud- SQL Databases
6. Google Cloud- NOSQL Databases
7. Amazon Databases

8. Oracle NoSQL Database

9. Cassandra DB

10. Data Center Engineering

11. Google File System (GFS)

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 find One, 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

AI Tool

MLSQL: An SQL-based language for machine learning; part of Apache Spark ecosystem. AI + SQL integration Case Study:

1. PostgreSQL
2. Oracle
3. IBM DB2 Universal Database
4. Microsoft SQL Server
5. SQLite database

Blog

1.OLAP tools from Microsoft Corp. and SAP
2.Views in database

3. Dynamic SQL and Embedded SQL
4. Active databases and Triggers'
5. SQL injection attack

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

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.

Course Outcome :

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

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	1	1				1		1	1	1	2
CO2	2	3	3	1	1								1	2
CO3	2	3	3	2	2			1			1	1	1	2
CO4	1	2	2	1	1					1		1	1	2
CO5	2	2	1	2	1	1			1	2	1		1	2
CO6	2	2	2	1	2	1		1	1			1	1	2
Average	1.8	2.5	2.3	1.3	1.3	1		1	1	1.5	1	1	1	2

FF No:654

AI 2304 : Discrete Structures and Computational Theory

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

Credits: 02 **Teaching Scheme Theory:02**Hours/Week

Course Relevance:

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

SECTION-I

Unit I Set Theory and Logic **(05** 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, Application of Propositional Logic Translating English Sentences, 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 **(04** 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 and Warshall's algorithm. Functions- Surjective, Injective and Bijective functions, Identity function, Partial function, Invertible function, Constant function, Inverse functions and Compositions of functions, The Pigeonhole Principle. (06 Hours)

#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 **(05** 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) **(04** 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) **(05** 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 **(05** 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

Text Books: (As per IEEE format)

3. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman,"Introduction to Automata Theory Languages and Computation", Addison-Wesley, ISBN 0-201-44124-1

4. 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. Systeam Programming

Job Mapping:

Job opportunities that one can get after learning this course

1. Research Scientist
2. Compiler Designer / 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	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
Average	2.66	2.5	2.66	1.88	2.16	0.16			0.16		1.16		2.66	1.33

FF No:654

AIM001: Augmented Reality and Virtual Reality

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

Credits: 03

Teaching Scheme Theory:02 Hours/Week

Tut: 01 Hours/Week

Course Relevance:

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

SECTION-I

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

Immersive Basics, taxonomy, method sand 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, Key point 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:

1. Understanding Immersive Technology and Its Basics
2. Real-World Applications and Consequences of AR/VR/MR
3. Augmented Reality vs. Virtual Reality – A Conceptual Comparison
4. VR Development Lifecycle and Comparison with SDLC
5. Collaborative Virtual Reality – Concepts and Benefits
6. Unity and Unreal Engine Templates and Programming Basics
7. Basics of Image Processing in AR – Feature and Keypoint Detection
8. Role of Generative AI in Virtual and Augmented Reality Content Creation
9. Generative AI for Real-Time Avatar and Scene Generation in Immersive Systems
10. Ethics, Security, and Challenges of Generative AI in AR/VR/MR

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: (As per IEEE format)

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

Moocs Links and additional reading material:

1.https://www.deepar.ai/demos

2.https://app.vectary.com/

3.https://builtin.com/articles/what-is-webar

Course Outcomes:

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

1. Describe the key concepts and applications AR/VR/MR.
2. Describe the Virtual Reality concepts clearly.
3. Create application development using Virtual Reality.
4. Create and demonstrate development using Augmented Reality
5. Develop and deploy 3D AR/VR models
6. Understand advances in Mixed Reality.

CO - PO Mapping:

	Program Outcomes (PO)												PSO	
CO/PO	**PO 1**	**PO 2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO2**
CO1	1										1	1		
CO2			2											
CO3			3		2	1							2	3
CO4	1													
CO5			3		2	1							2	3
CO6								2						
Average	0.5		1.33		0.6	0.3		0.3			0.16	0.16	0.6	

FF No:654

AIM002: Image Processing

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.

Credits: 03 **Teaching Scheme Theory: 02 Hours/Week**

Tut: 01Hours/Week

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:

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

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, Spatial domain techniques Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average, Image Smoothening: low-pass spatial filters, median filtering, Image Sharpening: high-pass spatial filter, derivative filters, Frequency domain techniques- Ideal low-pass filter, Butterworth low-pass filter, High-pass filter, Homo-morphic filters.

Unit 3: Image Analysis **(5 Hours)**

Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation Feature Extraction- Boundary representation(Chain code, B-spline representation, Fourier descriptor) Region representation (Area, Euler number, Eccentricity, Shape matrix, moment based descriptor), texture based features.

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.

Unit 6: Image Transform **(5 Hours)**

Introduction to two dimensional orthogonal and unitary transforms, properties of unitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine, Slant, KL, affine transforms. Singular Value Decomposition, Applications of transforms in Image processing.

List of Practical's: (Any Six)

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.

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:

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.

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	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

AIM 003: Computer Graphics

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.

Total Credits :03

Teaching Scheme:

Theory 02 Hours/Week

Tut:01Hours/Week

Course Relevance:

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

SECTION-I

Unit 1: **Introduction to Computer Graphics-** **(5 Hours)**

Definition, history and scope of computer graphics, pixel, Pixel dimensions, Typical graphics pipeline (CPU → GPU) and rendering equation, **Graphics file formats** – raster (BMP, PNG, JPEG, GIF, TIFF), vector (SVG), 3-D (OBJ, FBX, glTF), compression basics, Framebuffer concept, Color Models and Pixel Color ,(RGB, CMYK, HSL, HSV color models) 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 rasterization** – scan-line algorithm, inside-outside tests, edge tables, Area**-fill algorithms** – boundary-fill, flood-fill, seed-fill variants, **Primitive attributes** –intensity mapping point/line styles, area patterns**, Anti-aliasing** – super-sampling, area sampling, Wu's algorithm, gamma correction basics.

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 – Sutherland-Hodgman.

SECTION-II

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

3-D Cartesian & homogeneous coordinate systems; right-hand rule, 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, Composite 3-D transformations; Euler angles & gimbal lock, 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, shadow volumes (idea only)

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 Practicals:

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.
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
11. AI-based Handwriting Recognition and Rendering

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*
3. *WebGL,* ***8th Edition****, Pearson,* ***2020****.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,* ***3rd 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***.

Steven Harrington. ***Computer Graphics: A Programming Approach, 2nd Edition,*** *McGraw-Hill*

Moocs Links and additional reading material:

https://onlinecourses.nptel.ac.in/noc20_cs90/preview

Course Outcomes:

Upon completion of the course, student 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.

CO - PO Mapping:

	Program Outcomes (PO)											**PSO**	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	3	2	2	2	2	2	1			1		3	3
CO2	3	3	3	2	2					1		3	3
CO3	3	3	3	2	2					1		3	2
CO4	3	3	3	2	3		1			1		3	2
CO5	3	2	3	2	3		1			1		3	2
CO6	3	2	3	2	3					2		3	3
Average	2.0	1.5	2.3	1.7	1.3	1.8	0.0	0.0	0.0	0.0	0.0	2.3	2.5

FF No. : 654

HS2002: From Campus To Corporate – 1

Credits:.2 **Teaching Scheme**: Lab: 2 hours/Week

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,

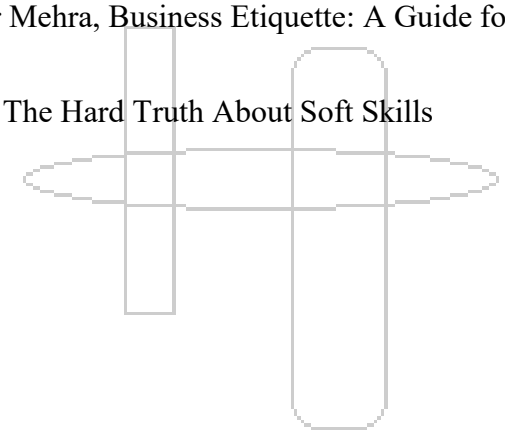
Introduction to Project Management Basics: Understanding tasks, milestones, deadlines, Collaboration using tools like Trello, Slack, Teams, Basics of Agile/Scrum concepts, Reporting and escalation protocol

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, HS2002: Reasoning and Aptitude Development - 3

Credits:.1 **Tut**: 1 Hour/Week

Unit 1: English Language

Familiarity with English Language, Ability to understand written text, spoken word and effective communication through written documents; Coverage of vocabulary to cope up with general and specific terminology, syntax and sentence structure, prevention of incorrect use leading to distortion in communication; synonyms, antonyms and contextual vocabulary, Grammar – Error identification, sentence improvement and construction, Reading Comprehension

Unit 2: Logical Ability

Objective interpretation of things, ability to perceive and interpret trends to make generalizations; ability to analyze assumptions behind an argument or statement; Deductive reasoning: Assessment of ability to synthesize information and derive conclusions - Coding deduction logic, Data Sufficiency, Directional Sense, Logical word sequence, Objective reasoning, Selection and decision tables, puzzles; Inductive reasoning: Assessment of ability to learn by example, imitation or by trial – Analogy pattern recognition, Classification pattern recognition, Coding pattern recognition, Number series pattern recognition; Abductive reasoning: Critical thinking ability of seeing through logical weak links or loopholes in an argument or a group of statements; Critical reasoning: assessment of ability to think through and analyze logical arguments, assessment of ability to use logical constructs to offer reasoning in unfamiliar situations; Information Gathering and synthesis: Ability of locating information, information ordering, rule based selection and data interpretation, order and classify data, interpret graphs, charts, tables and make rule based deductions. Application of these approaches for using visual, numerical and textual data from single or multiple sources

Unit 3: Quantitative Ability

Basic numbers – decimals and fractions, factorization, divisibility: HCF, LCM, Odd, even, prime and rational numbers. Application of algebra to real world, direct and inverse proportion, common applications – Speed-time -distance, Profit-loss, percentage, age relations, mixtures, other miscellaneous quantitative combination, exponentials and logarithms, permutations and combinations, probability. Spatial reasoning: Inductive – Missing portions, Sequence and series; Deductive analysis.

Reference Books –

1. "English Grammar in Use" by Raymond Murphy, Cambridge University Press.
2. "Word Power Made Easy" by Norman Lewis, Goyal Publishers & Distributors.
3. "Objective General English" by S.P. Bakshi, Arihant Publications.
4. "English for Competitive Examinations" by K. Sinha, S. Chand Publishing.
5. "Essential English Grammar" by Philip Gucker, Wiley.
6. "English Idioms and Phrasal Verbs" by M.A. Yadav, Vikas Publishing House.
7. "The Oxford English Grammar" by Sidney Greenbaum, Oxford University Press.
8. "A Modern Approach to Verbal & Non-Verbal Reasoning" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978-8121903409.
9. "Logical Reasoning and Data Interpretation for the CAT" by Nishit K. Sinha, Pearson India, ISBN: 978- 8131709117.
10. "Logical Reasoning and Data Interpretation for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
11. "A New Approach to Reasoning Verbal and Non-Verbal" by B.S. Sijwali & Indu Sijwali, Arihant Publications, ISBN: 978-9311124692.
12. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal, S. Chand Publishing, ISBN: 978- 8121900637.
13. "How to Prepare for Quantitative Aptitude for the CAT" by Arun Sharma, McGraw Hill Education, ISBN: 978-0070709642.
14. "The Pearson Guide to Quantitative Aptitude for Competitive Examination" by Pearson, Pearson India, ISBN: 978-8131709117.
15. "Quantitative Aptitude for Competitive Examinations" by Abhijit Guha, Tata McGraw Hill Education, ISBN: 978-0070666653.
16. "Data Interpretation & Data Sufficiency" by R.S. Aggarwal, S. Chand Publishing ISBN: 978-8121903515.
17. "Quantitative Aptitude for Competitive Examinations" by S. Chand, S. Chand Publishing, ISBN: 978- 8121903423.

Course Outcomes:

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

1. Improve the reading, writing and verbal skills, and enhance comprehension and articulation abilities
2. Develop logical reasoning abilities, enabling them to make sound decisions in problem-solving scenarios
3. Develop mathematical aptitude as well as data interpretation abilities and use them in test cases and real world problems
4. Learn to apply approaches for optimum time-management, prioritization maximizing the accuracy
5. Learn data interpretation, apply mathematical skills to draw accurate conclusions
6. Apply their knowledge of English, reasoning and quantitative skills for planning, critical thinking and real world problems

7. **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	2	2					3		2	2		
CO2	2	2	3	2	2		2		3		2	2		
CO3	2	2	3	2	3		2		3		2	2	2	
CO4	2	2	3	2	3	3		2	3		2	2		
CO5	2	2	3	2	3	2			3		2	2	1	
CO6	2	2	3	3	2				3		3	2	1	
Average	2	1.83	2.83	2.83	2.6	2.5	1.66	3	3	0	2.16	2	0.6	

FF No.: 654

AI2305: Design Thinking I

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

Credits: 01 **Tut:** 01 Tut Hours/Week

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**	**PO12**	**PSO1**	**PSO2**
CO1	2	1	2	2	1	3	1	3	2	2	2	1	3	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	3	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

AI 2306: Engineering Design & Innovation I

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

Credits: 04 **Teaching Scheme Theory: 00** Hours/Week

Tut: 00 Hours/Week

Lab: 08 Hours/Week

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)												**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

AI2307: Object Oriented Programming

Course Prerequisites:

1. Fundamentals of Computer programming and problem solving.

Course Objectives:

1. To introduce the fundamental concepts and structure of C++ and its compilation process.
2. To develop problem-solving skills using control structures, functions, and basic data types in C++.
3. To demonstrate the principles of Object-Oriented Programming such as encapsulation, abstraction, inheritance, and polymorphism.
4. To implement and analyze the use of constructors, destructors, operator overloading, and friend functions.
5. To explore advanced features of C++ including inheritance, polymorphism, dynamic memory management, and exception handling.
6. To apply templates and use the Standard Template Library (STL) for efficient generic and reusable programming.

Credits: 04

Teaching Scheme Theory: 02Hours/Week

Tut: 00 Hours/Week

Lab: 02 Hours/Week

Course Relevance:

The course "Object-Oriented Programming using C++" 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: Introduction to C++, Structure of a C++ Program, Compilation Process, Basics of OOP: Class, Object, Abstraction, Encapsulation, Inheritance, Polymorphism. **(03 Hours)**.

Basic Programming in C++: Data Types, Operators, Control Structures (if, switch, loops), Functions in C++: Function declaration, definition, call, default arguments, inline functions.

(04 Hours)

Classes and Objects: Defining Classes and Creating Objects. Access Specifiers, Member Functions, Scope Resolution Operator, Static Data Members. **(03 Hours)**

Constructors and Destructors: Constructors: Types (Default, Parameterized, Copy), Constructor Overloading. Destructors and its Role in Memory Management. **(02 Hours)**

Operator Overloading and Friend Functions: Operator Overloading: Concept, Overloading Unary and Binary Operators. Friend Function and Friend Class: Need and Implementation.

(02 Hours)

SECTION-II

Inheritance: Types of Inheritance (Single, Multilevel, Multiple). Hierarchical and Hybrid Inheritance, Constructor Behavior in Inheritance. **(3 Hours)**

Polymorphism: Function Overloading and Function Overriding, Virtual Functions and Runtime Polymorphism. (**3 Hours)**

Pointers and Dynamic Memory: Pointers: Basics, Pointer to Object, this Pointer. Dynamic Memory Allocation (new/delete), Array of Objects Using Pointers. **(3 Hours)**

Exception Handling: Basics of Exception Handling: try, catch, throw. Exception Handling with Multiple Catch Blocks, Custom Exceptions. **(3 Hours)**

Templates and STL: Function Templates and Class Templates. Introduction to Standard Template Library (STL): Vectors, Iterators. **(2 Hours)**

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 Practical's: (Any Six)

1. Develop a simple AI chatbot using classes and basic OOP concepts in C++.
2. Implement a rule-based decision system for health advisory using control structures.
3. Write modular C++ functions for basic AI data preprocessing and normalization.
4. Create a Neuron class to simulate an artificial neuron using objects and static members.
5. Demonstrate use of constructors and destructors in initializing AI model parameters.
6. Overload operators in a Vector class to perform basic AI vector operations.
7. Implement inheritance to design a hierarchy of AI model types.
8. Demonstrate function overloading and runtime polymorphism in AI evaluation metrics.
9. Use pointers and dynamic memory allocation to simulate AI dataset loading.
10. Handle exceptions during AI model training using try-catch blocks and custom exceptions.
11. Design generic AI components using function and class templates in C++.
12. Use STL vectors and iterators to manage and process AI datasets.
13. Generate creative sentences by combining random words from arrays of nouns, verbs, and adjectives.

14. Develop a basic chatbot in C++ using rule-based logic, simulating how GenAI gives intelligent replies.

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:

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

List of Course Group Discussion Topics:

1. Is C++ Still Relevant for AI and Machine Learning Development?
2. Object-Oriented Programming vs Functional Programming in AI Systems
3. The Role of Inheritance and Polymorphism in Building Scalable AI Models
4. Can Operator Overloading Enhance Code Readability in AI Applications?
5. Should AI Systems Rely on Manual Memory Management in C++?
6. Exception Handling: A Must-Have for Reliable AI Applications?
7. Templates and Generic Programming: Are They Suitable for AI Frameworks?
8. Using Standard Template Library (STL) in Real-Time AI Data Processin
9. The Impact of Constructor and Destructor Design on AI Performance
10. Dynamic Memory Allocation vs Static Allocation in AI Model Training
11. Friend Functions: Useful Feature or a Breach of Encapsulation in AI Codebases?
12. Is Python Replacing C++ in AI—Or Do They Serve Different Purposes?
13. Importance of Control Structures and Logic Building in AI Program Flow

14. Hybrid Inheritance in AI System Design: Too Complex or Highly Modular?

15. Balancing Performance and Abstraction in AI Development with C++

List of Home Assignments:

Blog:

1. Single and Multidimensional arrays in Java
2. Comparison Inheritance & Polymorphism
3. Need of abstract classes and interfaces in Java
4. Multithreading concept in Java
5. Signed & Unsigned arithmetic operations usin JAVA
6. Role of start() and run() methods in multithreading

Survey:

1. Strategies for Migration from C++ to Java
2. Product development using Inheritance and Polymorphism in Industry
3. on Java/OOP features popular amongst developers
4. Which other (non-JVM) languages does your application use?
5. How Java Impacted the Internet
6. How can a ArrayList be synchronised without using vector?

Design:

1. Implementation of Singleton design pattern in Java
2. Notes Repository System for Academic
3. Design for employee management system
4. Design for student management system
5. Inventory Management System
6. Write a program to delete duplicate numbers from the file

Case Study:

1. Java development milestones from 1.0 to 16.0
2. Implementation of Different Methods in Polymorphism
3. Real world systems which use java for its implementation
4. Drawing a flag using java
5. Use of different methods of Class object
6. Drawing a flag using java

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. Serra, C++ for Machine Learning, 1st ed. Birmingham, UK: Packt Publishing, 2020.*

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

Moocs Links and additional reading material:

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/#

Course Outcomes:

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

1. Understand the structure, syntax, and compilation process of C++ programs.
2. Write modular and efficient C++ programs using data types, operators, control structures, and user-defined functions.
3. Apply the principles of Object-Oriented Programming to model and solve real-world problems using classes and objects.
4. Implement constructors, destructors, operator overloading, and friend functions in C++ programs.
5. Analyze and apply different types of inheritance and polymorphism to develop extendable and reusable code.
6. Utilize templates and the Standard Template Library (STL) for generic programming and apply exception handling to develop robust applications.

Future Courses Mapping:

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

Job Mapping:

Job opportunities that one can get after learning this course

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	PO12	PSO1	PSO2
CO1	3	1			2									
CO2	3	2	2	1	2								2	
CO3	3	3	3	2	2				1			1	2	1
CO4	3	2	2	2	2				1					1
CO5	3	3	3	2	3				2			2	3	2
CO6	3	2	2	3	3				2	1	1	3	3	3
Average	3	2..16	2.4	2	2.33				1.5	1	1	2	2.5	2.33

FF No.: 654

AI 2308: COMPUTER NETWORK

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.

Credits: 03

Teaching Scheme:

Theory: 2 Hours/Week

Lab: 2Hours/Week

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.

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

SECTION-I

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.

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, piggybacking. Random Access Techniques: CSMA, CSMA/CD, CSMA/CA, Controlled Access Techniques: Reservation, Polling, Token Passing

Transport Layer (5)

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

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

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.

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

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 the sliding window protocol.
    - Go Back N
    - Selective Repeat
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 Framing methods.

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.

List of Course Seminar Topics:

1. Asynchronous Transfer Mode
2. Need Of Multiplexing for Signal Modulation
3. TDM with PAM a case study
4. Noise signal
5. Basic Network Protocols
6. Manchester Vs Differential Manchester coding technique
7. Amplitude Shift Keying: Working and Applications
8. Nyquist Sampling Theorem
9. CDMA
10. Line coding Techniques with example

List of Course Group Discussion Topics:

1. TCP/IP Model
2. Mobile IP
3. Congestion Control and QoS
4. Wireless Technology for Short range and long range
5. Application Protocols and its security
6. IP Protocols
7. Data Communication Issues in IP Networks and Solutions to it
8. Congestion control in hybrid networks
9. Issues in Real time Audio and video transmission protocol.
10. IPV6

List of Home Assignments:

1. Design the procedure to configure TCP/IP network layer services.
2. Simulation of Routing Protocols using NS2
3. Simulation of FTP based Protocols using CISCO packet Tracer/ NS2
4. Simulation of Congestion Control Protocols Using NS2

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. C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols", Prentice Hall, 2004

Moocs Links and additional reading material:

1. https://nptel.ac.in/courses/106105183
2. https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/
3. https://www.my-mooc.com/en/categorie/computer-networking

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. Analyze data flow between peer-to-peer in an IP network using Application, Transport and Network Layer Protocols.

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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	1	3	2	3	2	-	-	-	-	-	-	1	2
CO2	2	3	3	3	2	2	-	-	-	-	-	1	2
CO3	3	3	3	2	2	2	-	-	-	-	-	1	3
CO4	2	2	3	-	3	-	-	-	2	1	-	2	3
CO5	3	2	3	-	3	-	-	-	2	1	-	2	2
CO6	3	3	2	2	2	-	-	-	-	-	-	2	2
Average	2.3	2.3	2.6	1.6	2.3	0.6	-	-	0.6	0.3	-	1.5	2.3

FF No.: 654

AI 2309: Mathematical Foundation of Artificial Intelligence

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.

Credits: 3 **Teaching Scheme Theory**: 2 Hours/Week

Lab: 2 Hours/Week

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

Vectors and Matrix Operations (5 Hrs)

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.

System of Linear Equations, Eigen Value and Eigen Vectors (5 Hrs)

Systems of Linear Equations - Solving linear systems using Gaussian elimination and matrix inversion, rank and consistency of solutions, introduction to LU decomposition.
Eigenvalues and Eigenvectors - Concept of eigenvalues/eigenvectors, computation using characteristic polynomial, diagonalization and spectral theorem, applications in stability analysis.

First Order Logic (4 Hrs)

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,

SECTION-II

Inference in First-Order Logic (5 Hrs)

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

Propositional Logic (4 Hrs)

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.

Probabilistic Reasoning (5 Hrs)

Representing knowledge in uncertain domains, semantics of Bayesian networks, representation of conditional distributions, exact inference in Bayesian networks, variable elimination algorithm, approximate inference in Bayesian networks, relational and first-order probability models,

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 (PO)											PSO	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	1	1	0	0	0	0	0	0	0	0		0	0
CO2	1	1	0	0	0	0	0	0	0	0	0		1	0
CO3	1	1	2	0	0	0	0	0	0	0	0		0	1
CO4	0	1	0	2	0	0	0	0	0	0	0		0	0
CO5	1	1	1	0	1	0	0	0	0	0	0		1	0
CO6	2	2	1	1	0	0	0	0	0	0	0		0	1
Average	1.66	1.66	0.833	0.5	0.16	0	0	0					0.33	0.33

FF No.: 654

AI2310: PROBABILITY AND APPLIED STATISTICS

Course Prerequisites:

1.Basics of Mathematics

Course Objectives:

1. To facilitate the students with a concrete foundation of probability and statistics.
2. To analyze problems in Science and Engineering applications through probability and statistics methods.

Credits:02 **Teaching Scheme: Theory:02**Hours/Week

Course Relevance:

The course is relevant to all branches of Engineering. Its an important foundation for computer science fields such as machine learning, artificial intelligence, computer graphics, randomized algorithms, image processing, and scientific simulations.

SECTION-I

UNIT 1: Probability Theory 04 Hours.

Sample Space and Events, Classical, Empirical and Axiomatic Probabilitiies, Addition and Multiplication Theorems, Conditional Probability and Independence, Bayes' Theorem and Applications .

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, Addition Theorem of Expectation, Multiplication Theorem of Expectation, Continuous Distributions: Normal Distribution, Joint Probability Distribution, Discrete Distributions: Binomial distribution, Poisson's distribution.

UNIT III: Probability Densities 04 Hours

The Uniform Distribution, Log-normal Distribution, Beta Distribution, Gamma Distribution.

SECTION-II

UNIT IV: Descriptive Statistics and Data Presentation 06 Hours

Introduction to Statistics and its applications, Types of Data: Qualitative vs. Quantitative, Data Collection Methods and Sampling Techniques, Frequency Distribution and Data Tabulation, Graphical Representation of Data: Bar Chart, Histogram, Pie Chart, Box Plot, Measures of Central Tendency: Mean, Median, Mode, Measures of Dispersion: Range, Variance, Standard Deviation, Coefficient of Variation, Measures of Shape: Skewness and Kurtosis.

UNIT V: Sampling and Estimation 04 Hours

Sampling Techniques: Random, Stratified, Systematic, Cluster Sampling, Sampling Distribution of Sample Mean and Proportion, Central Limit Theorem, Estimation: Point Estimation and Interval Estimation, Confidence Intervals for Means and Proportions.

UNIT VI: 04 Hours

Correlation, Regression, and Non-Parametric Methods, Correlation Analysis: Pearson's and Spearman's coefficients, Simple and Multiple Linear Regression, Model Diagnostics: R^2, adjusted R^2, Residual Analysis, Introduction to non-parametric tests: Sign test, Wilcoxon rank-sum test, Kruskal–Wallis test.

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. *S. C. Gupta and V. K. Kapoor, Fundamentals of Applied Statistics, New Delhi, India: Sultan Chand & Sons, 2020.*
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. *M. R. Spiegel and L. J. Stephens, Schaum's Outline of Statistics, New York, NY: McGraw-Hill, 2008.*
5. *T. H. Wonnacott and R. J. Wonnacott, Introductory Statistics for Business and Economics, Hoboken, NJ: Wiley, 1990.*
6. *W. Mendenhall, R. J. Beaver, and B. M. Beaver, Introduction to Probability and Statistics, Boston, MA: Brooks/Cole, 2013.*

Course Outcomes:

Upon completion of the course, 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.

Future Courses Mapping:
Machine Learning, Statistical Theory and Interference.

Job Mapping:
Job opportunities that one can get after learning this course
1. Data Analyst
2. Data Scientist
3. Business Analyst

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	2	2						2	1	2	2
CO2	2	3	3	2	2						2	1	2	2
CO3	2	3	3	2	2						2	1	2	2
CO4	2	2	2	2	2						2	1	2	2
CO5	2	2	2	2	2						2	1	2	2
CO6	2	2	2	2	2						2	1	2	2
Average	2	2.5	2.5	2	2						2	1	2	2

FF No: 654

MM0102: Augmented Reality and Virtual Reality

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

Credits: 3

Teaching Scheme:

Theory:03 Hours/Week

Tut: 01 Hours/Week

Course Relevance:

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

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:

1. Understanding Immersive Technology and Its Basics
2. Real-World Applications and Consequences of AR/VR/MR
3. Augmented Reality vs. Virtual Reality – A Conceptual Comparison
4. VR Development Lifecycle and Comparison with SDLC
5. Collaborative Virtual Reality – Concepts and Benefits
6. Unity and Unreal Engine Templates and Programming Basics
7. Basics of Image Processing in AR – Feature and Keypoint Detection
8. Role of Generative AI in Virtual and Augmented Reality Content Creation
9. Generative AI for Real-Time Avatar and Scene Generation in Immersive Systems
10. Ethics, Security, and Challenges of Generative AI in AR/VR/MR

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, 2016s*

Reference Books: (As per IEEE format)

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.*
3. *Example - H. Schmidt-Walter, R. Kories; 'Electrical Engineering. A Pocket Reference'; Artech House, 2007. Accessed: Oct. 16, 2016. [Online]. Available: https://ebookcentral.proquest.com*

Moocs Links and additional reading material:

1. https://www.deepar.ai/demos
2. https://app.vectary.com/

https://builtin.com/articles/what-is-webar

Course Outcomes:

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

1. Describe the key concepts and applications AR/VR/MR.
2. Describe the Virtual Reality concepts clearly.
3. Create application development using Virtual Reality.
4. Create and demonstrate development using Augmented Reality
5. Develop and deploy 3D AR/VR models
6. Understand advances in Mixed Reality.

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**										**1**	**1**		
CO2			**2**											
CO3			**3**		**2**	**1**							**2**	**3**
CO4	**1**													
CO5			**3**		**2**	**1**							**2**	**3**
CO6								**2**						
Average	**0.33**		**1.3**		**0.66**	**0.33**		**0.33**			**0.166**	**0.166**	**0.66**	**1**

MM0103: Image Processing

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.

Credits: 03

Teaching Scheme:

Theory:02 Hours/Week

Tut: 01 Hours/Week

Course Relevance:

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

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, Spatial domain techniques { Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average, Image Smoothening: low-pass spatial filters, median filtering, Image Sharpening: high-pass spatial filter, derivative filters, Frequency domain techniques- Ideal low-pass filter, Butterworth low-pass filter, High-pass filter, Homo-morphic filters.

Unit 3: Image Analysis **(5 Hours)**

Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation Feature Extraction- Boundary representation(Chain code, B-spline representation, Fourier descriptor) Region representation (Area, Euler number, Eccentricity, Shape matrix, moment based descriptor), texture based features.

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.

Unit 6: Image Transform **(5 Hours)**

Introduction to two dimensional orthogonal and unitary transforms, properties of unitary transforms. One-two dimensional discrete Fourier Transform (DFT). Cosine, Slant, KL, affine transforms. Singular Value Decomposition, Applications of transforms in Image processing.

List of Practical's: (Any Six)

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 Tuatorials:

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.

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:

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.

CO - PO Mapping:

		ProgramOutcomes(PO)PSO												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**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 0104: Computer Graphics

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.

Teaching Scheme: 03

Teaching Scheme:

Theory: 2 Hours / Week;

Tut : 01 Hours / Week

Course Relevance:

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

SECTION-I

Unit 1: **Introduction to Computer Graphics**- (5 Hours)

Definition, history and scope of computer graphics, pixel, Pixel dimensions,. Typical graphics pipeline (CPU → GPU) and rendering equation, **Graphics file formats** – raster (BMP, PNG, JPEG, GIF, TIFF), vector (SVG), 3-D (OBJ, FBX, glTF), compression basics, Framebuffer concept, Color Models and Pixel Color ,(RGB, CMYK, HSL,HSVcolor models) 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, seed-fill variants, **Primitive attributes** –intensity mapping, point/line styles, area patterns**, Anti-aliasing** – super-sampling, area sampling, Wu's algorithm, gamma correction basics.

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 – Sutherland-Hodgman,

SECTION-II

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

3-D Cartesian & homogeneous coordinate systems; right-hand rule, 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, Composite 3-D transformations; Euler angles & gimbal lock, 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, shadow volumes (idea only)

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 Practicals:

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.
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
11. AI-based Handwriting Recognition and Rendering

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,* ***3rd 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*

Moocs Links and additional reading material:
https://onlinecourses.nptel.ac.in/noc20_cs90/preview

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

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. Create simple animations and explore graphics file formats and shaders

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	2	1			1			2	2
CO2	3	3	3	2	2					1			2	1
CO3	3	3	3	2	2					1		3	1	2
CO4	3	3	3	2	3		1			1			2	2
CO5	3	2	3	2	3		1			1			1	2
CO6	3	2	3	2	3					2		1	3	1
Average	2.0	1.5	2.3	1.7	1.3	1.8	0.0	0.0	0.0	0.0	0.0	0.5	1.8	1.6

FF No.: 654

AI2311: Design Thinking II

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

Credits: 01 **Tut:** 01 Tut Hours/Week

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	1	1	2	3	2
CO2	1	2	1	1	0	1	1	3	1	2	0	2	3
CO3	2	2	2	2	2	2	2	3	2	1	2	2	3
CO4	2	2	1	1	2	2	1	3	1	2	1	3	3
CO5	2	2	2	2	2	2	2	3	1	2	1	3	2
CO6	2	2	2	2	2	2	2	3	2	0	2	3	2
Average	2	2	1	2	2	2	2	3	2	2	2	3	3

FF No. : 654

AI 2312: Engineering Design & Innovation II

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

Credits: 04

Teaching Scheme:

Theory:00s Hours/Week

Tut: 00 Hours/Week

Lab: 04 Hours/Week

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 (PO)											**PSO**	
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

TY . B. Tech. Artificial Intelligence and Data Science

AY 2025-26

FF No.: 654

AI3001: Artificial Intelligence

Course Prerequisites:

1. A course on "Computer Programming and Data Structures"
2. A course on "Mathematical Foundations of Computer Science"
3. Some background in linear algebra, data structures and algorithms, and probability will be helpful

Course Objectives:

1. To learn the distinction between optimal reasoning Vs. human like reasoning
2. To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
3. To learn different knowledge representation techniques.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.

Credits: 4

Teaching Scheme Theory: 2 Hours/Week

Tut: 1 Hour/Week

Lab: 2 Hours/Week

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

Topic: Fundamentals of Artificial Intelligence (4 Hours)

Introduction: A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, Criteria for Success Turing Test. **Intelligent Agents**: Agents and Environments, concept of rationality, the nature of environments, structure of agents, types of agents, problem solving agents. problem formulation. **Formulation of problems**: Vacuum world, 8 queens, Route finding, robot navigation.[CO1, CO2] [PO1, PO2]

Topic: Uninformed Search Strategies (5 Hours)

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

Topic: Informed Search Methods: (5 Hours)

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

SECTION-II

Unit-IV Title: Logical Agents:

Knowledge based agents, Wumpus world. **Propositional Logic**: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. **First order Logic**: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. Representation, Structure, Backtracking. **Expert System**: Design, Implementation, Case study of Expert System in PROLOG

Unit-V Title: Natural Language Processing: Classic problems in NLP, Basic NLP Techniques, Applications: Information Retrieval, Concept Based Information Retrieval.[CO4] [PO2]

Unit-VI Title: Planning:

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]

List of Tutorials:

1. AI problem formulation [CO1] [PO1]
2. Task Environment [CO1] [PO1]
3. AI Problem Characteristics [CO1] [PO1]
4. Missionaries and Cannibals Problem [CO2] [PO2]

5. Water Jug Problem [CO2] [PO2]
6. Monkey Banana problem [CO2] [PO2]
7. 8 Puzzle Problem [CO2] [PO2]
8. Magic Square problem [CO2] [PO2]
9. Tic-Tac Toe Problem [CO3] [PO3]
10. Robot Navigation [CO5] [PO4]
11. Propositional Logic Examples [CO5] [PO4]
12. Predicate Logic Examples [CO5] [PO4]
13. Mini Expert system examples [CO5] [PO4]

List of Practical's: (Any Six)

1. Implementation of AI and Non-AI technique by implementing any two player game [CO1,CO2] [PO1,PO2]
2. Implementation of Uninformed strategies [CO1,CO2] [PO1,PO2]
3. Implementation of Informed strategies[CO2,CO3] [PO2,PO3]
4. Implementation of CSP Problem [CO3] [PO3]
5. Implementation predicate logic using PROLOG[CO5] [PO4]
6. Implementation of Expert system using PROLOG[CO5] [PO4]

List of Course Projects:

1. Inventory management E Commerce [CO1] [PO1]
2. stock market price prediction [CO1] [PO1]
3. Object Identification / detection [CO2] [PO2]
4. Product Delivery Drones [CO3] [PO3]
5. Pick and drop robotic arm [CO4] [PO2]
6. Arrangement of blocks [CO2] [PO2]
7. Smart city water / light management system [CO2] [PO2]
8. Human Tracking system [CO2] [PO2]
9. Automatic Interview Conduction system [CO3] [PO3]
10. Student Information Chatbot Project. [CO3] [PO3]
11. Product Review Analysis For Genuine Rating. [CO3] [PO3]
12. Customer Targeted E-Commerce [CO4] [PO2]
13. College Enquiry Chat Bot [CO2] [PO2]
14. Artificial Intelligence HealthCare Chatbot System [CO3] [PO3]
15. Intelligent Tourist System Project [CO3] [PO3]

List of Course Seminar Topics:

1. Fundamentals of Artificial Intelligence [CO1] [PO1]
2. Intelligent Agents [CO1] [PO1]

3. Uninformed searching Techniques [CO2] [PO2]
4. Informed searching Techniques [CO2] [PO2]
5. Gaming Techniques [CO2] [PO2]
6. Planning Techniques [CO5] [PO4]
7. Applications of AI [CO6] [PO3]
8. Predicate Logic [CO4] [PO2]
9. Propositional Logic [CO4] [PO2]
10. Adversarial Search Techniques [CO4] [PO2]

List of Course Group Discussion Topics:

1. Not Applicable to this Course

List of Home Assignments:

1. Not Applicable to this Course

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. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley,1990.

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

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

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

5. Carl Townsend, "Introduction to turbo Prolog", Paperback, 1987

Moocs Links and additional reading material:

1. *www.nptelvideos.in*

Course Outcomes:

Upon completion of the course, 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. Formulate and solve a given problem using Propositional and First order logic.
5. Analyze the AI problem using different planning techniques.
6. Design and carry out an empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports.

Future Courses Mapping:

1. *Machine Learning*

Job Mapping:

Job opportunities that one can get after learning this course

1. *AI Data Analyst, Data Scientist*

CO - PO Mapping:

	Program Outcomes (PO)											**PSO**	
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
CO1	2												
CO2		2											
CO3			2									1	
CO4		1											
CO5				3									
CO6			1										
Average	0.33	0.5	0.5	0.5								0.16	

FF No.: 654

AI 3002: Operating System

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

Credits: 04

Teaching Scheme Theory: 02 /Week

Tut: 01 /Week

Lab: **01** /Week

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) **4 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) 6 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) 4 Hours

Types of Scheduling algorithms FCFS, SJF, RR, Priority

SECTION-II

Unit 4: (Deadlock) 4 Hours

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

Unit 5: (Memory Management) 5 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) 5 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, Recod Blocking, Secondary Storage Management

List of Practical's: (Any Six)

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 Course Projects:

1. Design and implementation of a Multiprogramming Operating System: Stage I
    i. CPU/ Machine Simulation
    ii. 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

    I/O Handling, Spooling and Buffering

List of Course Seminar Topics:

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

    Protection and Security in Operating System

List of Course 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
   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

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" ; 9thEdition; 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.*

Moocs Links and additional reading material:

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

Course Outcomes:

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

Future Courses Mapping:

1. Advance Operating System
2. Unix Operating System
3. Linux programming
4. 4. Distributed System/Computing System Programming

Job Mapping:

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	1	3			3	2							3	
CO2	2	3		2	2			2			2			2
CO3	2	2		3	2									
CO4		3	3		3	2		2						
CO5	2	2							2					2
CO6							2	2			2	3		
Average	2.25	2.6	3	2.5	2.5	2	2	2	2		2	3	3	2

FF No.: 654

AI 3003 : Statistical Inference

Course Prerequisites: Basic knowledge of Statistics and Probability, Python

Course Objectives:

1. Get basic understanding about statistical models and their use.
2. Apply regression models with hyper-parameter tuning methods depending upon the problem context .
3. Get a better understanding of probabilistic models.
4. Derive inference from different statistical data sets.

Credits: 4

Teaching Scheme Theory: 2 Hours/Week

Tut: 1 Hours/Week

Lab: 2 Hours/Week

Course Relevance: Machine learning, Data Science

SECTION-I

Topics and Contents (5Hrs): Introduction to statistical inference, Purpose and Applications of Models, Practical Use of Models, key steps in the modeling process, Hypothesis Testing. Optimization Strategies in Parameter Estimation, Dimensionality reduction techniques, Factor analysis, Concept of Outliers

Correlation, Regression and Generalization (4Hrs): Correlation and its type, Correlation Analysis, Regression Analysis, Generalization and Model Evaluation

Regression Types and Generalization types(5hrs): Multivariate Linear Regression, Regularized Regression - Ridge Regression and Lasso, poisson regression, Theory of Generalization: Training and Testing Curves, Case Study of Polynomial Curve Fitting, Cross validation, Modelling non linear relationship

SECTION-II

Topics and Contents (5hrs): Introduction to probabilistic models, some examples of probabilistic models, Communication Models in Probabilistic Systems, Monte Carlo Simulation

Building blocks of probability models (5hrs), Key Probability Distribution (Bernoulli, Binomial, Normal distribution), mixture models, Bayesian method, Bootstrapping and Maximum Likelihood Methods, expectation maximization

Markov-chain models (4Hrs), Introduction to Markov Chains, Hidden Markov model, Conditional random fields, Latent variable probability models

List of Tutorials:

List of Tutorials:(Any Three)

1. Consider the following set of points:{(-2,-1),(1,1),(3,2)}

a. Find the least square regression line

For the given data points.

b. Plot the given points and the regression line in the same rectangular system of axes.

2. Find the Standard Deviation,Variance,Mean,Median,Modeforthefollowingdata7,11,1

1,15,20,20,28.

3. A2-Ddatasetisgivenbelow.

4. C1=X1={(4,1),(2,4),(2,3),(3,6),(4,4)}

5. C2=X2={(9,10),(6,8),(9,5),(8,7),(10,8)}

6. Calculate the dimensionality reduction using linear discriminant analysis.

1. Find the coefficient of Regression for the following

dataX12 3 4 5 6 7 8 9

Y9 8 10 12 11 13 14 16 15

2. Find whether Null-Hypothesis is correct or not using One-Way ANNOVAABC

23 4

45 6

67 8

6. Solve Poisson Regression model problem using a workable example.

Find the Principal Components for Z1, Z2 for the following matrixA

T= 2 1 0 -1

4 3 1 0.5

8. ADieisthrown6-times.If getting an odd number is a success what is the probability of

i. 5-Success

ii. Atleast 5-Success

iii. Atmost 5-Success

9. If a fair coin istossed10 timesthen find the probability of

i. Exactly 6 heads

ii. Atleast 6 heads

iii. Atmost 6 heads

10. In a bolt factory, Machines A, B and C manufacture respectively 25%, 35% and 40% of the total bolts. Out of their total output 5, 4 and 2 percentage are respectively defective bolts. A bolt is drawn at random from the product. If the bolt is defective, what is the probability that the Bolt is manufactured by Machine B.

List of Practicals: (Any Six)

1. Use any Generative AI tool to generate a synthetic dataset of at least 50 samples and implement regression model using least square estimation

2. Implementing Ridge and Lasso Regression using AutoML Tools

3. To evaluate the performance of a machine learning regression model using K-Fold Cross-Validation and interpret its generalization ability.

4. Exploratory Factor Analysis on Multivariate Data using SPSS

5. To understand and apply Principal Component Analysis (PCA) using Orange Data Mining for dimensionality reduction and visualizing patterns in multivariate data

6. Exploring Error Correction in Text using Noisy Channel Model vs. Hugging Face Transformers

7. Modeling the Source-Channel Communication Framework using Generative and Transformer-based Tools

8. Implement and Analyze the Maximum Likelihood Estimation (MLE) Technique using Synthetic and Real-world Datasets

9. To Understand and Implement the Expectation-Maximization (EM) Algorithm for Clustering and Parameter Estimation

10. To Model and Simulate Real-World Processes using Markov Chains

11. Sequence Modeling using Hidden Markov Models with Pomegranate

List of Projects:

1. To apply optimization and outlier detection on real-time data.

2. Use PCA and factor analysis to simplify features in a marketing dataset to reduce dimensions of customer behavior data, interpret components, and visualize clusters.

3. Apply Ridge and Lasso regression to prevent overfitting, use a dataset like Boston Housing to implement and compare models.

4. Implement Logistic regression to do credit score prediction using German credits score dataset and perform cross validation.

5. Implement factor analysis to find the important features out of all features present in the Student Performance Dataset.

6. Implement Principal Component analysis to identify the crucial features out Of all features present in the Breast cancer dataset.

7. Evaluation of Generalization Techniques in AI Models: Cross Validation, Bootstrap, and Regularization

8. Perform comparision analysis using various models

9. Probabilistic Simulation of Communication Systems using Monte Carlo Techniques

10. Implement different feature selection techniques on any data set.

11. Sequence Modeling with Hidden Markov Models for Activity Recognition

12. Conditional Random Fields for Named Entity Recognition in Text

List of Course Seminar Topics: NA

List of Course Group Discussion Topics: NA

List of Home Assignments:

Design:

1. Heart disease prediction

2. Customer Review classification

3. Sensorless drive diagnosis

4. Default credit card client classification

5. Devnagri handwritten character classification

6. Biased sample

7. Kappa statistics

8. Entropy estimation

Case Study:

1. Statistical Inference in the Age of AI: Concepts, Challenges, and Applications

2. Classification models

3. Bayesian Inference in Machine Learning: Probabilistic Thinking for AI Models

4. Regression models

5. Maximum likelihood

6. Generalized linear discriminant analysis.

7. Conditional Randomfields

8. Monte carlo simulation

9. Maximum Likelihood Estimation (MLE): Foundation of Modern Model Fitting

Blog

1.Logistic regression

2.Support vector machine

3. Types of error

4. Markov chain model

5. Latent variable probability model

6. Conditional random fields

7. Dimensionality Reduction: PCA, Factor Analysis, and Their Role in AI

8. Linear discriminantanalysis

9. Anova

10. Ancova

11. Poisson Regression

12. Entropy estimation

13. Monte Carlo Simulation in AI: Approximating the Intractable

14. Expectation Maximization Algorithm and Its Role in Unsupervised Learning

15. The Bias-Variance Tradeoff: Finding the Sweet Spot in AI Models

16. Bootstrapping Techniques in Model Validation and Ensemble Learning

Surveys

1. Random forest vs Decision tree

2. Principal Component analysis

3. Bayesian method

4. Types of distribution

5. Different variance models

Text Books: (As per IEEE format)

1.The Elements of Statistical Learning: Data Mining, Inference, and Prediction. By Trevor Hastie, Robert Tibshirani, Jerome Friedman, Hardcover: 745 pages, Publisher: Springer; 2nded.2009,ISBN- 10:0387848576

2.Statistical Models by A.C.Davison Paperback:738pages,Publisher:Cambridge University Press;1edition(30June2008)ISBN- 10:0521734495CambridgeUniversityPress

3. An Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani

4. The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman

Reference Books: (As per IEEE format)

1. S.C.Gupta;"Fundamentals of Statistics 7th Edition"; Himalaya Publishing House Pvt.Ltd.

2.Abdul Hamid Khan, MANOJ KUMAR SRIVASTAVA, and NAMITA SRIVASTAVA;"STATISTICAL INFE RENCE:THEORYOF ESTIMATION";PhiLearning

3. Bayesian Reasoning and Machine Learning" by David Barber

Moocs Links and additional reading material:

1.Statistics tutorial-https://www.youtube.com/channel/UCQKwruq0LY3cjvSx7_M5JAg

2. Statistical Inference: https://www.classcentral.com/course/statinference-1717

Course Outcomes:

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

1. Understand and Explain Statistical methods used in data modeling.
2. Analyze various correlation methods to provide insights from different datasets.
3. Apply appropriate regression and optimization techniques to evaluate model performance.
4. Formulate given problem using probabilistic approch for concise and accurate analysis.
5. Interpret various distribution methods for effective model construction.

6. Apply Markov modeling process to efficiently compute and evaluate probabilistic models.

Future Courses Mapping:

Machine learning, Deep Learning

Job Mapping:

Job opportunities that one can get after learning this course

For all jobs in the domain of AI&DS knowledge of statistical inference is prerequisite. To name a few Big Data Engineer, Business Intelligence Developer, Data Scientist, Machine Learning Engineer, Research Scientist, AI Data Analyst, Product Manager, AI Engineer, Robotics Scientist, Machine Learning Architect etc.

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								3	0
CO2	2	3		2	1								2	0
CO3	3	3	2										0	2
CO4	3	2	2	3	2						1	1	3	2
CO5	2	2		2	1								2	0
CO6	2	3	2										3	2
Average	2.5	2.5	1	1.33	1						0.6	0.6	2.16	1

AI 3004: Machine Learning

Course Prerequisites:

1. Linear Algebra, Statistics, Probability, Calculus, and Programming Languages

Course Objectives:

1. Explain and compare fundamental machine learning paradigms and differentiate among key algorithms
2. Apply core mathematical concepts linear algebra, probability, statistics, and optimization to formulate and solve machine learning problems.
3. Design and implement machine learning models, and evaluate their performance on real-world datasets.
4. Perform data preprocessing and feature engineering techniques to prepare raw data for modeling, and select appropriate methods to improve accuracy and robustness.
5. Analyze and mitigate issues like overfitting, underfitting, bias-variance trade-off, and assess model interpretability and scalability for deployment.

Credits: 4

Teaching Scheme Theory: 02 Hours/Week

Tut: 01 Hours/Week

Lab: 02 Hours/Week

Course Relevance:

Machine Learning is the applicable science of making computers work without being explicitly programmed. It is mainly an application of Artificial Intelligence (AI) that allows systems to learn and improve from experience, without any human intervention or assistance. Machine Learning keeps on innovating every aspect of the business and has been shaping up the future even more powerfully now. Machine learning is the fuel we need to power robots, alongside AI. With ML, we can power programs that can be easily updated and modified to adapt to new environments and tasks- to get things done quickly and efficiently. Machine learning skills help you expand avenues in your career.

SECTION-I

Types of Learning (6 hrs): 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 Classification.

Decision Tree Learning: Representation, Basic decision tree learning algorithm, Issues in decision tree learning, and Random Forest Model. Validation: Cross validation, Confusion matrix.

Bayesian Learning (5 hrs): Probability, Bayesian Learning: Bayes theorem, Naïve Bayes algorithm, Maximum likelihood hypothesis. Ensemble Learning: Bagging and boosting.

SVM (3 hrs): Kernel functions, Linear SVM, Nonlinear SVM, Hyper parameter tuning, Handling Imbalanced Data set. KNN Model.

SECTION-II

Clustering Algorithms (5 hrs) : Unsupervised learning, clustering. Partition based clustering, K-means and K Medoid, Hierarchical clustering, Density based clustering algorithms.

Association rules mining (5hrs) : Apriori Algorithm, Confidence and Support parameters. Introduction to Hidden Markov model, Genetic algorithm. Dimensionality Reduction Techniques: PCA, SVD etc.

Reinforcement learning (4 hrs): Exploration, Exploitation, Rewards, Penalties, Markov Decision Process, QLearning and Bellman Equation. Artificial Neural Networks: Basics of ANN, Feed Forward Neural Networks, Deep neural networks etc.

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: (Any Six)

1. Implement simple linear regression using a dataset.
2. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using Decision Tree.
3. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using Random Forest.
4. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using Naïve Bayes.
5. Implement Find-S algorithm.
6. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using SVM
7. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using KNN classifier.
8. Train the system using data set obtained from UCI ML repository. Use a partition of the same data set as a test set to determine accuracy using K-Means clustering
9. Implement the ANN algorithm on a data set obtained from UCI ML repository
10. Apply PCA and SVD on a data set obtained from UCI ML repository
11. Movie Review Sentiment Classifier with AI-Augmented Dataset

List of Course Projects:

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

1. Sentiment analysis of movie /restaurant dataset
2. Possibility of heart attack based on text data.
3. Market basket analysis
4. Credit Card Fraud Detection
5. Handwritten Digit Recognition
6. Image Caption Generator
7. Movie Recommendation System
8. Cancer Classification
9. Traffic Signs Recognition
10. Customer Segmentation using Machine Learning
11. Uber Data analysis

12. Loan prediction
13. HVAC needs forecasting
14. Customer relationship management
15. Clinical decision support systems
16. Fraud detection
17. Portfolio & Price Prediction
18. Smart Building Energy Management System
19. Quick analysis of quality of cereals, oilseeds and pulses
20. Building a Recurrent Neural Network
21. Operations on Word vectors
22. Neural Machine translation with attention

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

List of Course Group Discussion Topics:

1. Supervised Vs Unsupervised
2. Univariate Vs Multivariate analysis
3. Accuracy measuring methods
4. Bias Vs Variance Tradeoff
5. Data Reduction Vs Dimensionality reduction
6. Continuous Vs Discrete variables
7. Feature Extraction Vs Automatic Feature detection

List of Home 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

Text Books: (As per IEEE format)

1. *T. Mitchell, — Machine Learning, McGraw-Hill, 1997.*
2. *Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, Edition 2012*

Reference Books: (As per IEEE format)

1. *EthemAlpaydin, "Introduction to Machine Learning", MIT press, 2004.*
2. *"Data mining: concepts and techniques", Jiawei Han and Micheline Kamber the Morghan Kaufman, 2001.*
3. *J. Gabriel, Artificial Intelligence: Artificial Intelligence for Humans (Artificial Intelligence,*

Machine Learning), Create Space Independent Publishing Platform, First edition , 2016

Moocs Links and additional reading material:

1. https://www.coursera.org/learn/machine-learning
2. https://cs229.stanford.edu/
3. https://ocw.mit.edu/courses/6-036-introduction-to-machine-learning-fall-2020/
4. https://cs229.stanford.edu/syllabus.html
5. https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/
6. http://www.arxiv-sanity.com/
7. https://thegradient.pub/
8. https://distill.pub/

Course Outcomes:

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

1. Explain various types of machine learning paradigms and apply concept learning methods to determine the version space of hypotheses.

2. Apply Bayesian and ensemble learning techniques for classification and prediction tasks.

3. Apply classification algorithms such as decision trees, random forests, SVM, and KNN, and evaluate their performance using validation techniques.

4. Compare partition-based, hierarchical, and density-based clustering methods in the context of unsupervised learning.

5. Use association rule mining, dimensionality reduction, and intelligent algorithms to solve basic data analysis problems.

6. Classify key concepts of reinforcement learning and neural network models in the context of machine learning.

Future Courses Mapping:

1. Deep Learning

Job Mapping:

Job opportunities that one can get after learning this course

1. ML Engineer
2. Data Scientist
3. Data Analyst

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	0	0	0	0	0	0	0	0	1	2	0
CO2	3	2	3	2	3	2	0	1	2	1	0	1	0	2
CO3	3	3	3	2	3	2	0	0	2	1	0	0	2	0
CO4	3	2	0	2	2	0	0	0	2	1	0	0	2	0
CO5	3	2	2	2	3	2	0	1	0	0	0	1	0	2
CO6	2	2	2	2	2	0	0	0	2	1	0	1	0	2
Average	2.83	2.33	2.4	2	2.6	2	0	1	2	1	0	1	2	2

AI 3005: Engineering Design and Innovation-V

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.

Credits: 6 **Teaching Scheme**

Theory: 00 Hours/Week

Lab: 12 Hours/Week

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 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:

4. 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	0.83

FF No.: 654

AI 3013: Design Thinking V

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

Credits: 1 **Teaching Scheme Tut:** 1 Hour/Week

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

AI 3011: Complexity and Algorithms

Course Prerequisites:

1. 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.

Credits:4

Teaching Scheme Theory: 2 Hours/Week

Tut: 1 Hours/Week

Lab: **2** Hours/Week

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

Introduction: Algorithm analysis,Time complexity: Growth of Function: Asymptotic notation, Standard notations and common functions **Complexity analysis**-Time and space trade offs in algorithms(Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms., Recurrences. Methods for finding 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 algorithms 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 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 Practical's: (Any Six)

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 k^{th} 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 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:

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

List of Course Group Discussion Topics:

1. Greedy Algorithms Vs. Dynamic Programming strategy
2. Dynamic Programming Vs Greedy
3. NP-completeness
4. P Vs NP problems
5. Paradigms for algorithm design
6. Different Searching techniques
7. Relevance of Cook-Levin theorem
8. Randomness in computation

List of Home Assignments:

Can be of Design, Case Study, Blog, Survey

1. **Asymptotic Growth Rate Challenge** *List functions in increasing order of growth: e.g., log n, n, n log n, n^2, 2^n, n!, etc. Justify with reasoning.*
2. **Recurrence Relation Worksheet** *Solve and compare time complexities for recurrences using substitution, recursion tree, and Master Theorem.*
3. **Design and Analyze Your Own Sorting Algorithm** *Create a custom sorting method for specific types of data (e.g., nearly sorted, small-range integers). Analyze best, average, worst cases.*
4. **Greedy vs DP: The Coin Change Problem** *Implement both greedy and dynamic programming solutions. Identify test cases where greedy fails.*
5. **Travel Route Planner (Graph Assignment)** *Given a weighted map of cities, implement Dijkstra's and Bellman-Ford. Compare output on graphs with negative weights.*
6. **Build a Dependency Resolver using Topological Sort** *Model a system (e.g., course prerequisites or software installation) and implement topological sorting.*
7. **Explore NP-Complete Problems** *Pick a problem (e.g., Subset Sum, 3-SAT, Graph Coloring), describe why it's NP-Complete, and propose an approximate or heuristic solution.*
8. **Create a Visualizer for Recursion or DP** *Use any language or tool (e.g., Python, JavaScript) to animate recursion trees or DP tables for problems like Fibonacci, LCS.*
9. **Randomized Algorithms in Action** *Implement and analyze performance of randomized QuickSort vs regular QuickSort. Submit both timing results and correctness verification.*
10. **Graph Coloring for Timetable Scheduling** *Model a real-world scenario (e.g., exam scheduling) using graph coloring. Implement greedy coloring algorithm.*
11. **Compare Brute Force, Backtracking, and DP for Subset Sum** *Write all three approaches. Use input data of increasing size and compare execution times.*
12. **Algorithm Efficiency Battle (Class Leaderboard)** *Give a hard problem (e.g., finding longest increasing subsequence) and ask students to submit optimized code. Rank by execution time on large test cases.*

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 McGraw-HillEdition.*
2. *2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI.*
3. *3. John Kleinberg, Eva Tardos, "Algorithm Design", Pearson.*
4. *4. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design", Wiley Publication*

Moocs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

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

1. Understand the basic notation for analyzing the performance of the algorithms.
2. To apply appropriate algorithmic paradigms to design efficient algorithms for computational problems
3. To apply suitable mathematical techniques to analyze the asymptotic complexity of the algorithm for more complex computational problems.
4. To understand the significance of NP-completeness of some decision problems and its relationship with the tractability of the decision problems.
5. To understand the significance of randomness, and approximability in computation and design randomized and approximation algorithms for suitable problems
6. To incorporate appropriate data structures, and algorithmic paradigms to craft innovative scientific solutions for complex computing problems

Future Courses Mapping:

1. Advanced Algorithms
2. Graph Theory and Network Algorithms
3. Parallel and Distributed Algorithms

Job Mapping:

Job opportunities that one can get after learning this course

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	0	0	0	0	1	2	0	3	3	0
CO2	3	3	3	2	1	1	1	0	0	0	2	2	0	2
CO3	2	2	3	3	2	0	0	0	0	2	0	2	0	0
CO4	3	2	0	0	0	0	0	0	0	0	0	2	0	0
CO5	3	2	3	0	0	2	0	0	3	0	0	0	1	3
CO6	0	3	2	3	3	0	3	0	0	3	0	0	0	2
Average	2.8	2.3	2.8	2.2	2	1.5	2	0	2	2.3	2	2.25	2	2.3

FF No.: 654

AI 3012: Software Design and Methodologies

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

Credits: 4

Teaching Scheme Theory : 02 Hours/Week

Tut : 01 Hour/Week

Lab : 02 Hours/Week

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. **(4 Hours)**

Software Modeling : Introduction to Software Modeling, Advantages of modeling, Principles of modeling. **(2 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. **(4 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 **(2 Hours)**

Study of classes (analysis level and design level classes)

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). **(2 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. **(2 Hours)**

Interaction Diagram :

Sequence diagram, Interaction Overview diagram, State machine diagram, Advanced State Machine diagram, Communication diagram, Timing diagram. **(3 Hours)**

Architecture in the Life Cycle :

Architectural styles, Architecture in Agile Projects, Architecture and Requirements, Designing an Architecture. **(2 Hours)**

Design Patterns :

Introduction, Different approaches to select Design Patterns, Creational patterns, Structural Patterns, Behavioral Patterns **(3 Hours)**

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 Practical's: (Any Six)

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. To identify Entity,Control,Boundary objects and trace object interactions for scenarios from use cases.
7. Prepare a state chart diagram for a given object scenario.
8. To prepare detailed Activity diagram with notational compliance to UML2.0 indicating clear use 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 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

List of Course Group Discussion Topics:

1. Traditional Vs Agile
2. Phases of SDLC. Which is more important?
3. UML modeling
4. Analysis Vs Design

5. Design Patterns
6. Design Vs Architecture
7. Architecture style
8. Design Vs Framework
9. Framework Vs Architecture
10. Archetype patterns

List of Home Assignments:

Design:

1. Requirement Engg steps
2. Analysis modeling
3. Design modeling
4. Architectural styles
5. Design patterns

Case Study:

1. Imaging Software architecture
2. Banking Software architecture
3. ERP Software architecture
4. Online Shopping Software architecture
5. AI Software architecture

Blog:

1. Software Engg Do's and Don'ts
2. Which Process Model?
3. Scrum
4. Devops
5. Data ops

Surveys:

1. Software Design
2. Software Methodology

3. Software Architectures

4. Design Patterns

5. Architectural Patterns

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

Moocs Links and additional reading material:

www.nptelvideos.in

Course Outcomes:

Upon completion of the course, 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.

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:

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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**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

AI 3010: Deep Learning

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

Credits: 4 **Teaching Scheme Theory:** 2 Hours/Week

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

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 (4 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 (4 Hours): Regularization and optimizations, hyper parameter tuning, batch normalization, Dropout, data augmentation, Deep learning frameworks, Over fitting concepts, Exploding and Vanishing Gradient Problem

Convolutional Neural Networks (6 Hours) : Convolutional Neural Networks, padding, strided convolution, pooling layers, convolutional implementation of sliding windows, ResNet, Transfer Learning

SECTION-II

Sequence Models (6 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 (4 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 (4 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 (Generalised 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 sequential Modelling?
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. Illustrate logistic regression model, structured as a shallow Neural network.
2. Build and train a deep Neural Network
3. Apply techniques to improve neural network performance
4. Demonstrate understanding of functionality of all layers in a convolutional neural network
5. Understand and Apply Architecture of Generative Adversarial Networks
6. Demonstrate Understanding of Recurrent nets and their applications

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**
CO1	2				**2**									2
CO2	1								2		2		2	
CO3	2								1				1	2
CO4		2							2					1
CO5	1		2		2						**3**			
CO6		2	2		2						3	2		
Average	2				**2**									2

AI3008 - Engineering Design & Innovation VI

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.

Credits: 4

Teaching Scheme :

Theory: 0 Hours/Week

Lab: 12 Hours/Week

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 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 i 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:

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	
	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

AI 3015 : Design Thinking VI

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

Credits: 1 **Teaching Scheme**

Tut: 1 Hour/Week

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

B. Tech. Final Year Artificial Intelligence And Data Science

AY 2025-26 Module VII and VIII Course Content

AI 4025 :: High Performance Computing

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

Credits: 2 **Teaching Scheme**

Theory: 2 Hours/Week

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

AI4015: NETWORK SECURITY

Course Prerequisites: Computer Networks

Course Objectives:

Credits: 2 **Teaching Scheme Theory:** 2 Hours/Week

Course Relevance:

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

Data Science is a multidisciplinary field. It uses scientific approaches, procedures, algorithms and frameworks to extract knowledge and insight from a huge amount of data.

Data Science uses concepts and methods which belong to fields like information technology, Mathematics, Statistics, Computer Science etc.

Data Science influences the growth and improvements of the product by providing a lot of intelligence about customers and operations, by using methods such as data mining and data analysis.

The course is relevant to all branches of Engineering and beyond, since data is generated as an obvious outcome of many processes.

Unit 1 **(5Hours)**

Introduction

Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, Avoiding attacks, Security services.

key security properties - Confidentiality, Integrity, Availability.

Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack, Dictionary Attacks.

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: **(4 Hours)**

Private key cryptography

athematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), Role of random numbers in security, Importance of prime number, DES, AES.

Chinese remainder theorem

Unit 3: **(5 Hours)**

Public key cryptography

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

Unit 4 **(5 Hours)**

Authentication and access control

Message authentication and Hash Function. Authentication: One-Way Authentication, Mutual Authentication, 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

Part A: 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:

Cyber Attack, Cyber Reconnaissance, Crimes in Cyber Space-Global Trends & classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digital forensic

Text Books

1. "Cryptography and Network Security-Principles and Practices" by William Stallings, Pearson Education, 2006, ISBN 81-7758-774-9, 4th Edition

2. "Network Security and Cryptography", by Bernard Menezes, Cengage Learning, 2010, ISBN 81-315-1349-1, 1st Edition

Reference Books

1. "Computer Security: Art and Science", by Matt Bishop, Pearson Education, 2002, ISBN 0201440997, 1st Edition.

2. "Network security, private communication in a public world", by Charlie Kaufman, Radia Perlman and Mike Spencer, Prentice Hall, 2002, ISBN 9780130460196, 2nd Edition

3. "Cryptography and Information Security", by V.K. Pachghare, PHI, 2015, ISBN-978-81-203-5082-3, Second Edition

Additional Reading

1."Security architecture, design deployment and operations", by Christopher M. King, Curtis Patton and RSA press, McGraw-Hill, 2001, ISBN 0072133856, 1st Edition.
2 'Inside Network Perimeter Security" by Stephen Northcott, Leny Zeltser, et al, Pearson Education Asia, ISBN 8178087618, 1st Edition.

Course Outcomes

Upon completion of the course, the students 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 7 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.

AI 4007: REINFORCEMENT LEARNING

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.

Credits:2 **Teaching Scheme Theory:**2Hours/Week

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
The Reinforcement Learning Problem: Reinforcement Learning, Examples, Elements of Reinforcement Learning, Limitations and Scope 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 Dynamic Programming: Policy Evaluation, Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration, Efficiency of Dynamic

Programming

Model-free solution techniques: Temporal difference learning, Monte Carlo Methods, Efficient Exploration and value updating

SECTION-2

Topics and Contents

Batch Reinforcement Learning: Introduction, Batch Reinforcement Learning Problem, Foundations of Batch RL Algorithms, Batch RL Algorithms, Batch RL in Practice

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

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

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 VIVA 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
6) Explain various tabular solution methods.

FF No.:654

AI4005: MAJOR PROJECT

Credits: 10 **Teaching Scheme Lab:** 20 hours/week

Course Relevance:

This is a culmination of four years of learning into Practical. This course is essential for Graduate Engineers to practice the successful management of a software development project. The course emphasizes on project life cycle phases requirement engineering, system analysis and system design and gives them the exposure to research in any area of their interest. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and It is a way of increasing the student's maturity and preparing him/her for their future career. The students carry out cutting edge projects with a flexibility to balance between research- and application-oriented work as per their interest. The program enables the students to find opportunities for higher studies in top ranking universities abroad, and to find jobs in dream companies .

The Motivation for this Major Project is

a. Synthesis of knowledge
b. To demonstrate the aptitude of applying the own knowledge to solve a specific problem.
c. To mature the knowledge.
d. Preparation for joining the working world.

The Project Work can lead to:

a. Novice algorithm development
b. Optimization of existing system/method
c. New state of the art application
d. Some incremental work in any existing field of their choice

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.

3. The project must be open-ended – meaning that there is not a known correct answer to the design problem. Students are expected to apply their creativity (simply copying or re-creating something that already exists is not acceptable).
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The
5. experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
6. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting Requirement Definition Document, Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
7. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.
8. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.
9. The Student Project Groups are expected to work on the recommendations given by the panel of examiners. In no case any variation in Project Theme will be permitted.
10. The outcome of the project should be tangible in terms of paper publication/patent/SOP/prototype
11. The Project should justify the work worth 10 credits.

Assessment Scheme

Sr. No.	Content	Marks
1	Development of Prototype/ Model	20
2	Innovativeness and intellectual input	20
3	evaluation of literature review	10
4	Individual contribution	10
5	Usage of Modern Tool/ Technology and experimental competency	10
6	Presentation of the Project Work	10
7	Results and analysis	10
8	Quality Publication and Project Report	10

Note:

The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice like signal processing, computer vision, machine learning and artificial intelligence, control systems, game theory, and communication networksand address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE

students. The Project work will be jointly performed by the project team members.
The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.
Following is the list of recommended domains for Project Work:
signal processing, computer vision, machine learning and artificial intelligence, IoT, Block Chain, Image Processing, data Science etc.

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Model the Real World Problem
2. Identify the Design within Specification and Available Resources
3. Realize the Solution within Defined references
4. Defend his Design with Technical and Ethical reasoning
5. Adapt to changing Technological and Human resource advances
6. Use the gained knowledge for other Real-World Problems
7. Project will involve development of a compact solution to current problem/s in chosen field.

AI 4023 : Design Thinking VII

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

Credits: 1 **Teaching Scheme Tut:** 1 Hour/Week

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

AI4008: INDUSTRY INTERNSHIP

Credit: 16

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
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, andmodern 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**

AI4011: INTERNATIONAL INTERNSHIP

Credit: 16

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to inculcate research culture.

SECTION-1

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

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: 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**

AI4010: RESEARCH INTERNSHIP

Credit: 16

Course Relevance: Implementation of technical knowledge acquired during previous three years of Internship and to inculcate Industry culture.

SECTION-1

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

Suggest an assessment Scheme:

MSE review for 50 marks converted to 30

ESE review for 100 marks converted to 70

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**

AI4009: PROJECT INTERNSHIP

Credit: 16

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**