Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. (Computer Engineering)

With Effect from Academic Year 2026-27

Prepared by: - Board of Studies in Computer Engineering

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

Chairman – BOS **Chairman – Academic Board**

Vision of the Institution

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

Mission of the Institution

- 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.

Vision of the Department

"To be a leader in the world of computing education practising creativity and innovation".

Mission of the Department

- To ensure students' employability by developing aptitude, computing, soft, and entrepreneurial skills
- To enhance academic excellence through effective curriculum blended learning and comprehensive assessment with active participation of industry
- To cultivate research culture resulting in knowledge-base, quality publications, innovative products and patents
- To develop ethical consciousness among students for social and professional maturity to become responsible citizens

Program Outcomes [PO]

PO	Program Outcome Statements
PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

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 team work:** 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 these to 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.

Program Specific Outcomes (PSOs)

Program Specific Outcome Statements

PSO1 Select and incorporate appropriate computing theory principles, data structures and algorithms, programming paradigms to innovatively craft scientific solution addressing complex computing problems.

PSO2 Adapt to new frontiers of science, engineering and technology by getting acquainted with heterogeneous computing environments and platforms, computing hardware architectures and organizations through continuous experimentation.

PSO3 Conceive well-formed design specifications and constructs assimilating new design ideas and facts for identified real world problems using relevant development methodologies and practices, architecture styles and design patterns, modeling and simulation, and CASE tools.

PSO4 Exercise research and development aptitude focusing knowledge creation and dissemination through engineering artifacts construction, preparation and presentation of engineering evidences using procedures, techniques, guidelines, and standards considering technology migration and evolution.

Program Educational Objectives (PEOs)

- Demonstrate application of sound engineering foundations to be a committed technology workforce
- Apply mathematical and computing theory knowledge base to provide realistic computer engineering solutions
- Exhibit problem-solving skills and engineering practices to address problems faced by the industry with innovative methods, tools, and techniques
- Develop professional and ethical practices adopting effective guidelines to acquire desired soft skills in the societal and global context
- Aim for continuing education and entrepreneurship in emerging areas of computing

Course Name Nomenclature as per NEP (For FY and SY)

BSC: Basic Science Course	MDOE: Multi Disciplinary Open Elective
ESC: Engineering Science Course	CC: Co-curricular Course
PCC: Program Core Course	HSSM: Humanities Social Science and Management
PEC: Program Elective Course	IKS: Indian Knowledge System
ELC: Experiential Learning Course	FP: Field Project
MD: Multi Disciplinary	INT: Internship

Nomenclature for Teaching and Examination Assessment Scheme AY 2024-25

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	Computer Science	CS
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 **FF No. 653**

Branch: Computer **Year:** T.Y. **A.Y.:** 2026-27 **Semester: V**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme										Credits
			Th	Lab	Tut	CVV	CA	MSA	ESA						Total	
							LAB (%)	MSE (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	PRACT+ CVV (%)	ESE (%)		
S1	CS3212	PCC: Design and Analysis of Algorithm	3	2	0	20		-	-	10	30	-		40 TH(W)	100	4
S2	CS3213	PCC: Computer Networks	2	2	0								60	40 TH(W)	100	3
S3	CS3111	PEC1:Web Technologies	3	2	0	20		-	20	-	40	20	--	--	100	4
S3	CS3112	PEC1: Mainframe Technology	3	2	0	20		-	20	-	40	20	--	--	100	4
S3	CS3113	PEC1: Basics of Game Development	3	2	0	20		-	20	-	40	20	--	--	100	4
S4		PEC2: Coursera	0	0	0									100(FE-Final evaluation)	100	3
S5		Open Elective 1: Coursera	0	0	0									100(FE-Final evaluation)	100	4
S6	MDM***	MDM: Blockchain Technologies	2	0	1				10		30	20		40 TH(W)	100	3
S6	MDM***	MDM: Ethical Hacking	2	0	1				10		30	20		40 TH(W)	100	3

S6	MDM***	MDM: Augmented and Virtual Reality	2	0	1				10		30	20		40 TH(W)	100	3
S6	MDM***	MDM: Advanced Computer Architecture	2	0	1				10		30	20		40 TH(W)	100	3
S6	MDM***	MDM: Data Science	2	0	1				10		30	20		40 TH(W)	100	3
S7	CS3227	Design Thinking - 5	0	0	0									100 (Direct Evaluation)	100	1
S8	CS3225	Engineering Research And Innovation - I	0	0	0			30 (R)						70 (R)	100	2
		Total	10	6	1											**24**

R: Review Examination
W: Written Examination

PEC2 Courses*: Coursera

Subject Code	Subject Name	Subject Code	Subject Name
MD3141	AWS Cloud Technology Consultant	MD3106	Microsoft Power BI Data Analyst
MD3101	IBM Full Stack Software Developer	MD3113	Google Data Analytics
MD3102	Meta Back-End Developer	MD3121	IBM DevOps and Software Engineering
MD 3112	Google Cyber Security		

Open Elective Courses:**

Subject Code	Subject Name	Subject Code	Subject Name
MD3201	MD3201:FINTECH	MD3203	MD3203:ENTREPRENEURSHIP
MD3202	MD3202:DIGITAL MARKETING	MD3204	MD3204:MANAGEMENT

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** T.Y. **A.Y.:** 2026-27 **Semester: VI**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme									Credits
			Th	Lab	Tut	CA	MSA	ESA						Total	
						LAB (%)	MSE (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	PCC	PCC: Artificial Intelligence and Machine Learning	3	2	0									100	4
S2	PCC	PCC: Cryptography and Network Security	2	2	0									100	3
S3	PEC1	PEC1: Compiler Design	3	2	0									100	4
S3	PEC1	PEC1: UI/UX Design	3	2	0										4
S3	PEC1	PEC1: PCGPU	3	2	0										4
S3	PEC1	PEC1: Image Processing	3	2	0										4
S4		PEC2 Courses*	0	0	0										3
S5		Open Elective Courses**	0	0	0									100	4
S6	MDM	MDM: Mathematics for Machine Learning	2	0	1										3
S6	MDM	MDM: Operational Research	2	0	1										3
S6	MDM	MDM: GenAI	2	0	1										3
S6	MDM	MDM: Randomized and Approximation Algorithms	2	0	1										3
S6	MDM	MDM: Business Intelligence	2	0	1										3

S6	MDM	MDM: Quantum Computing	2	0	1										3
S7		CS3141:Design Thinking - 6	0	0	0										1
S8		CS3151:Engineering Research And Innovation - II	0	0	0										2
		Total				**30**	**30**	**20**	**50**	**60**	**20**	**60**	**330**	**600**	**24**

PEC2 Courses*: Coursera

Subject Code	Subject Name	Subject Code	Subject Name
	MLOPs		Natural Language Processing
	DevOps		Information Retrieval

Open Elective Courses:**

Subject Code	Subject Name	Subject Code	Subject Name
OE	MD3201:FINTECH	OE	MD3203:ENTREPRENEURSHIP
OE	MD3202:DIGITAL MARKETING	OE	MD3204:MANAGEMENT

Semester-I

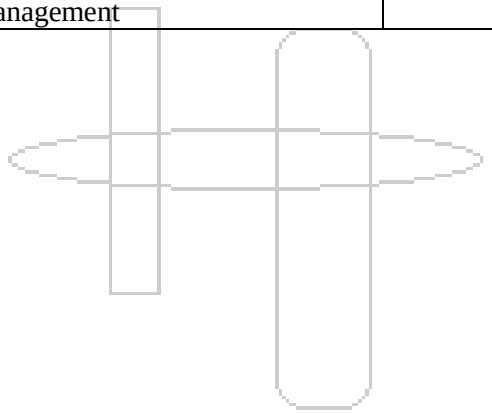
Title: Course Structure **FF No. 653**

Branch: Computer **Year:** BTech **A.Y.:** 2026-27 **Module: VIII (Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	LinkedIn Learning *	LinkedIn Learning	0							100			100	2
OE2	CS4217	Human Computer Interaction	2	-	-	10	-	30	-	30	-	30	100	2
	CS4209	Parallel Computing	2	-	-	10	-	30	-	30	-	30	100	2
	CS4222	Image Processing	2	-	-	10	-	30	-	30	-	30	100	2
OE3	CS4275 Swayam	Introduction to Machine Learning	2	-	-	10	-	30	-	30	-	30	100	2
	CS4276 Swayam	Deep Learning	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CS4225	Major Project	0	20	-	-	-	30	-	70	-	-	100	9
	CS4292	Design Thinking -7	-	-	-	-	-	-	-	100	-	-	100	1
		Total	**4**	**20**	-	**20**	-	**90**	-	**330**	-	**60**	**500**	**16**

LinkedIn Learning Courses*

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



Semester-I

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** BTech **A.Y.:** 2026-27 **Module: VIII (Internship - Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CS4236	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	15
S1	CS4238	Research Internship	-	32	-	-	-	30	-	70	-	-	100	15
S2	CS4292	Design Thinking-7	-	-	1	-	-	-	-	100	-	-	100	1
		Total		**32**	1	-	-	**30**	-	**170**	-		**200**	**16**

Semester-II

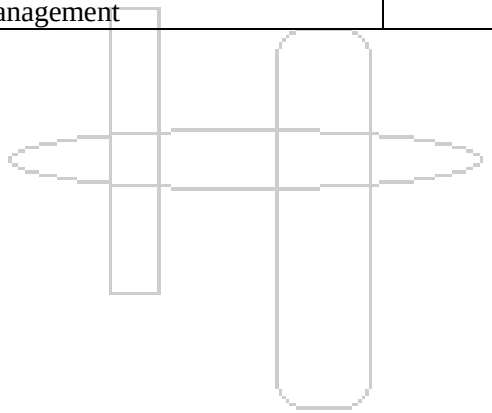
Title: Course Structure **FF No. 653**

Branch: Computer **Year:** BTech **A.Y.:** 2026-27 **Module: VIII (Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	LinkedIn Learning *	LinkedIn Learning	0							100			100	2
OE2	CS4217	Human Computer Interaction	2	-	-	10	-	30	-	30	-	30	100	2
	CS4209	Parallel Computing	2	-	-	10	-	30	-	30	-	30	100	2
	CS4222	Image Processing	2	-	-	10	-	30	-	30	-	30	100	2
OE3	CS4275 Swayam	Introduction to Machine Learning	2	-	-	10	-	30	-	30	-	30	100	2
	CS4276 Swayam	Deep Learning	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CS4226	Major Project	0	20	-	-	-	30	-	70	-	-	100	10
		Total	**4**	**20**	-	**20**	-	**90**	-	**230**	-	**60**	**400**	**16**

LinkedIn Learning Courses*

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



Semester-II

Title: Course Structure **FF No. 653**

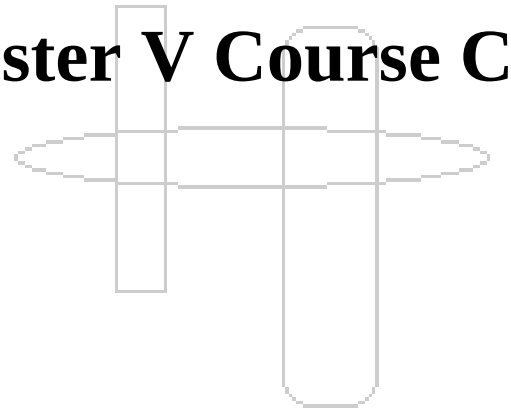
Branch: Computer **Year:** BTech **A.Y.:** 2026-27 **Module: VIII (Internship - Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CS4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	16
S1	CS4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	16
		Total		**32**	1	-	-	**30**	-	**170**	-		**200**	**16**

INDEX

T. Y. B. Tech. Computer Engineering
AY 2026-27

Semester V Course Content



Syllabus Template **FF No. :654**

CS3212:: Design and Analysis of Algorithms

Credits: 4

Teaching Scheme Theory: 3 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites: Basic courses on programming, data structures, discrete structures, theory of computing.

Course Objectives:

1. Students will gain understanding of asymptotic notations and will be able to apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms.
2. Students will develop the ability to formulate computational problems in the abstract and mathematically precise manner.
3. Student will gain understanding of different algorithm design paradigms such as divide and conquer, dynamic programming, greedy, backtracking and will apply suitable paradigm for designing algorithms for computational problems.
4. Students will develop understanding of notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems.
5. Students will gain understanding of basics of network flow optimization and max-flow min-cut theorem.
6. Students will be able to incorporate algorithm design principles, data structures and provide efficient solutions for complex computational problems.

Course Relevance:

This is a foundational course for Computer Science and Engineering. This course develops algorithmic thinking capability of students. Designing algorithms using suitable paradigms and analyzing the algorithms for computational problems has a high relevance in all domains where computer science plays a crucial role (equally in Industry as well as research). This course is also an essential pre-requisite for advanced domain specific algorithmic courses such as Algorithmic Graph Theory, Algorithmic Number Theory, Computational Geometry, Motion planning and Robotics, etc., to give a few examples. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates systematic study of any other domain (in computer science or otherwise) which demands logical thinking. This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, advanced algorithmic research.

Section 1: Topics/Contents

Unit-I Basic introduction and time and space complexity analysis (7 Hours)

Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms. Analysis of recursive and non-recursive algorithms.

Overview of basic algorithms: searching, sorting, graph traversals (DFS, BFS) and applications, topological sorting.

Introduction to adversary lower bound technique (for the comparison-based sorting algorithms), proving correctness of algorithms.

Unit-II Recursion and Divide and Conquer (9 Hours)

Divide and conquer strategy, role of recurrence relations in time complexity analysis, using recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications. Recursive strategies for enumeration of combinatorial objects (sets, permutations, combinations), Josephus Problem, Tower of Hanoi.

Application of divide and conquer for solution of computational problems: Quick sort , Merge sort, counting inversion, Finding a majority element, Order statistics (randomized and deterministic algorithms), closest pair of points in plane, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast modular exponentiation),

Introduction to randomness in computation, randomized algorithms for majority search, randomized quick sort (with analysis).

Unit-III Dynamic Programming (8 Hours)

Dynamic Programming strategy, Application of dynamic programming for solution of some computational problems: computing Fibonacci numbers, binomial coefficients, Matrix chain multiplication, Coin change problem, Traveling Salesperson Problem, Optimal Binary Search Tree construction, shortest paths in directed acyclic graphs, All pair shortest path algorithm, longest increasing subsequence problem, longest common subsequence, largest independent set for trees.

Section2:Topics/Contents

Unit-IV Greedy and Backtracking strategy (7 Hours)

Greedy: General strategy, Analysis and correctness proof of minimum spanning tree and shortest path algorithms, fractional knapsack problem, conflict free scheduling, Huffman coding.

Backtracking: General strategy, backtracking strategy for some problems: n-queen problem, graph coloring, Hamiltonian cycle problem, subset sum problem, solution of SUDOKU puzzle.

(Note: for all the strategies discussed in Units II, III, and IV the student is expected to understand the essence of the strategies and not just the specific applications and should be able to apply suitable strategies for fresh problems)

Unit-V Network Flows Optimization (4 Hours)

Flow in network, definition, basic properties, Ford Fulkerson Algorithm, cuts in graph, Max-flow Min-cut theorem, matchings in bipartite graphs.

Unit-VI Introduction to theory of NP-completeness (7 Hours)

Complexity classes P, NP, coNP, and their interrelation, Notion of polynomial time many one reductions reduction, Notion of NP-hardness and NP-completeness, Cook-Levin theorem and implication to P versus NP question, NP-hardness of halting problem. NP-Complete problems (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, clique, vertex cover problem, Integer Linear Programming (ILP))**.** modeling NP-complete problems using ILP.

List of Practical's:

1. Enumeration of combinatorial objects (sets, permutations, combinations)
2. Empirical computation of average number of max updates in standard max finding algorithm.
3. Josephus Problems (using arrays, linked list, recurrence relation, binary representation-based method)
4. Tiling Problem
5. Huffman coding
6. Optimal Binary Search tree construction
7. Application of dynamic programming for a two player game
8. SUDOKU solver
9. Modeling NP-complete problems using Integer Linear Programming (Vertex cover, independent sets, cliques)

List of Course Project topics:

1. Applications of A* algorithm in gaming
2. Pac-Man game
3. File compression techniques
4. Solution of Maze (comparing the backtracking based solution and Dijkstra's algorithm)
5. Different exact and approximation algorithms for Travelling-Sales-Person Problem
6. Creation of Maze using backtracking
7. Knight tour algorithms
8. Network flow optimization and maximum matching
9. AI for different games such as minesweeper, shooting games, Hex, connect-4, sokoban, etc
10. Graph theoretic algorithms
11. Computational Geometry Algorithms
12. AKS primality testing
13. Algorithms for factoring large integers.
14. Randomized algorithms for primality testing (Miller-Rabin, Solovay-Strassen)

15. Slider puzzle game

List of Course Seminar Topics:

1. Divide and Conquer Vs Dynamic Programming
2. Greedy strategy
3. NP-hardness
4. Backtracking strategy
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Philosophical relevance of P Vs NP question
8. Complexity classes
9. Space complexity
10. Compression Techniques
11. Real world applications of Graph theoretic algorithms
12. Approximation algorithms
13. Hardness of approximation
14. Pseudorandom number generators

Text Books: ***(As per IEEE format)***

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. New Delhi, India: Prentice-Hall of India, 2009.

[2] J. Kleinberg and E. Tardos, Algorithm Design, 1st ed. Boston, MA, USA: Pearson, 2005.

[3] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, Algorithms, 1st ed. New York, NY, USA: McGraw-Hill Education, 2006.

Reference Books

[4] R. Motwani and P. Raghavan, Randomized Algorithms, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1995.

MOOCs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

The student will be able to –

1. To formulate computational problems in abstract and mathematically precise manner
2. To design efficient algorithms for computational problems using appropriate algorithmic paradigm
3. To analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.
4. To establish NP-completeness of some decision problems, grasp the significance of the notion of NP-completeness and its relationship with intractability of the decision problems.
5. To understand basics of network flow and max-flow min-cut theorem
6. To incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solutions for complex computing problems.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	2	3	2	1			2					2	3		2	
2	2	3	3	1			2					2	3		2	
3	2	3	2	1			2					2	3		2	
4	2	3	3	1			2					2	3		2	
5	2	3	3	1			2					2	3		2	
6	2	3	3	1			2					2	3		2	
Avg	**2**	**3**	**2.66**	**1.0**			**2.0**					**2.0**	**3**		**2**	

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

Advanced Algorithms, Computational Complexity, Computational Geometry, Algorithmic Number Theory, Algorithmic Graph Theory

Job Mapping:

Algorithm design lie at heart of any Computer Science/Engineering application. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic studying any other domain (in computer science or otherwise) which demands logical thinking. Algorithm design is an essential component of any job based on programming. All Industries in computer Engineering always look for a strong knowledge in Algorithm design and Data structures. If student wants to pursue higher education/ research in Computer Science, this course is must.

Syllabus Template **FF No. : 654**

CS3213::Computer Networks

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

Lab: 2 Hours/Week

Course Prerequisites: Operating System, Theory of Computer Science

Course Objectives:

1. To learn the data communication model, signal generation, data encoding, digital modulation and demodulation required for wired and wireless communication networks.
2. To learn the physical layer which includes transmission mediums, physical layer devices, transmission modes and topologies, performance issues for intranet and internetworks.
3. To learn multiple access schemes and wide area network connectivity for intranet and internetworks.
4. To learn IP protocol and routing algorithms for packet switching service framework used in intranet and internetworks.
5. To learn TCP and UDP protocol to provide quality of service over packet switching service framework used for intranet and internetworks.
6. To learn to select, design, develop, analyze and evaluate client server solutions for societal requirements at large.

Course Relevance:

The key technology of the information age is communications. Data communications and networking is a truly global area of study, both because the technology enables global communication over telephone lines and Internet. Data communication and networking is the backbone of all IT infrastructures in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world.

Section 1: Topics/Contents

Unit-I Data Communication, Networking Fundamentals, Physical Layer 05 Hours

Communication Model: Source, Transmitter, Transmission System, Receiver, Destination, Data Terminal Equipment (DTE), Data Communication Equipment (DCE). Transmission Configurations: Point to Point and Multipoint. Transmission Modes: Synchronous and Asynchronous. Transmission Methods: Serial and Parallel. Communication Modes: Simplex, Half Duplex, and Full Duplex.

Digital Line Encoding: Unipolar NRZ, Polar NRZ, NRZ Inverted, Bipolar, Manchester, Differential Manchester and Pulse Code.

Networking Fundamentals: Types of Computer Networks: LAN, MAN, WAN, PAN, Internet, internet and Intranet. Network Architectures: Client-Server; Peer To Peer. Network Topologies: Mesh, Star and Hierarchical. Addressing: Physical &logical Addresses, Port Addresses, Specific Addresses. Connecting devices: Hubs (Passive, active, Intelligent), Switches (Layer-2, Layer-3 and Managed), Bridges, Routers, Gateway.
Reference Models: OSI, TCP/IP. Design Issues for Layers. Is ATM still used? Is ISDN dying? Is Frame Relay outdated? Is SNA still present in the Market?. SDN.

Unit-II Logical Link Control **05 Hours**

Logical Link Control: Design Issues: Services to Network Layer, Framing, Error Control: Parity Bits, Hamming Codes and CRC. Flow Control Protocols: Sliding Window Protocol, WAN Connectivity: PPP and HDLC. PPPoE, PPPoA. Is DOCSIS used today? Do we use DSL line today? Is PPP still used?

Unit-III Medium Access Control **04 Hours**

Medium Access Control: Channel Allocation: Static and Dynamic, Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA. Legacy Standard : 10 Mbps IEEE 802.3 Standard(Ethernet), Wiring Schemes and Frame Formats, CSMA/CD, Binary Exponential Back-off Algorithm. High Speed Ethernet Standards: Fast, Gigabit and 10Gigabit (Focus must be on Gigabit Networks). Wireless Standards, CSMA/CA.

Section2: Topics/Contents

Unit-IV Network Layer **05 Hours**

Network Layer: Switching Techniques: Circuit, Message and Packet Switching. Logical Addressing: IPv4 and IPv6, Subnetting, NAT, CIDR. Network Layer Protocols: ARP, IP, ICMP, Routing Protocols: Distance Vector, Link State. Routing in Internet: RIP, OSPF, BGP, Congestion control and QoS, MPLS, Mobile IP.

Unit-V Transport Layer **05 Hours**

Transport Layer: Services: Berkeley Sockets, Addressing, Connection Establishment, Connection Release, Flow control and Buffering, Multiplexing. HTH Layer Protocols: TCP, TCP Timer management, UDP. Quality of Service: TCP Congestion Control. Traffic Shaping: AIMD. QUIC Protocol.

Unit-VI Application Layer **04 Hours**

Application Layer: Domain Name System (DNS). WWW: Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0). Web Service. Email: SMTP, MIME, POP3 and Webmail. File Transfer: FTP, Dynamic Logical Addressing: Dynamic Host Control Protocol (DHCP).

List of Practical's (Minimum Six):

1) Write a program in C++/JAVA to implement - Unipolar NRZ, Polar NRZ, NRZ Inverted, Bipolar Encoding, Manchester Encoding and Differential Manchester Encoding.
2) Setting up small computer networks and Hands on networking commands:

A) Set up a small wired and wireless network of 2 to 4 computers using Hub/Switch/Access point. It includes installation of LAN Cards, Preparation of Cables/ Installation and Configuration of Access Point, Assigning unique IP addresses and use of ping utility. Hands on for network commands - ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap.

B) Switch and Router Configuration using Cisco Packet Tracer and Switch Port Assignment.

3) Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.(50% students will perform Hamming Code and others will perform CRC). Further extend it to real implementation of CRC over Ethernet standard.
4) Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode. Further extend it to real implementation of Flow Control over TCP protocol.
5) Dynamic Routing Deployment via Single-Area OSPFv2.
6) Write a program using TCP Berkeley socket primitives for wired /wireless network for following
a. Say Hello to Each other
b. File transfer
c. Calculator (Arithmetic)
d. Calculator (Trigonometry)
Demonstrate the packets captured traces using Wireshark/Fiddler for traffic analysis tool in peer to peer mode.
7) Write a program using UDP Berkeley Sockets for wired/wireless network to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark/Fiddler for traffic analysis tool in peer to peer mode.
8) Understanding protocol stack of Intranet
a) Analyze packet formats of Ethernet, IP, TCP and UDP captured using Wireshark/Fiddler for traffic analysis tool in peer to peer mode for wired and wireless networks.
b) Use any tool for custom packet generation (Packet Sender Tool) or write your own code for packet generation and analyze the packets.
9) Develop a client-server using C++ or JAVA to demonstrate the behavior of HTTP1.0, HTTP1.1, HTTP1.2 and HTTP2.0 protocols along with all success and error messages. Use Firefox as client browser.

List of Suggested Course Project areas (for reference):

1. Simulation of modulation and demodulation for digital telephone lines
2. Simulation of modulation and demodulation for 100 Mbps Ethernet Network
3. Simulation of modulation and demodulation for Gigabit Ethernet Network
4. Simulation of modulation and demodulation for 10Gigabit Ethernet Networks
5. Simulation of modulation and demodulation for 3G for mobile networks
6. Simulation of modulation and demodulation for 4G mobile networks
7. Develop a tool fox for line encoding methods
8. Develop a tool fox for modulation and demodulation methods
9. Design and deploy TCP based Multithreaded HTTP client server for accessing student activity data in the institute.
10. Design and deploy TCP based Multithreaded FTP client server to share institute level notices.
11. Design and deploy UDP based Multithreaded TFTP client server for your class
12. Design and deploy TCP based Multithreaded SMTP and POP3 mail client server for your campus.
13. Design and deploy TCP based Multithreaded Chat client server for your class.
14. Design and deploy UDP based Multithreaded Chat client server for your class.
15. Design and deploy UDP based Multithreaded Audio Conferencing client server for computer engineering department.
16. Design and deploy UDP based Multithreaded Video Conferencing client server for computer department
17. Implementation of RIP/OSPF/BGP using Packet Tracer
18. Simulation of AODV routing protocol using Packet Tracer/ NS3/OMNet

Text Books: ***(As per IEEE format)***

1. Andrew S. Tanenbaum, "Computer Networks",5th Edition, PHI, ISBN 81-203-2175-8.

2. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204

3. Frouzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006

Reference Books: ***(As per IEEE format)***

1. Matthew S. Gast "802.11 Wireless Networks", O'Reilly publications; 2nd Edition.

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

3. Holger Karl and Andreas Willing, "Protocols and Architectures for Wireless Sensor Networks", Wiley, ISBN: 0-470-09510-5

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com, CCNA certification.

Course Outcomes:

The student will be able to –

1. Select line encoding, modulation, topology, essential components of physical layer, data transmission rates to design computer networks.
2. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency.
3. Propose mechanisms for server channel allocation in wired and wireless computer networks.
4. Design and analyze IP-based networks using network layer protocols and other services.
5. Evaluate transport layer protocols and services for reliable end-to-end communication.
6. Compare sustainable engineering practices indicating the scientific purpose and utility of communication frameworks and standards at application layer.

CO-PO Mapping:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	**2**			**2**									3		
2	3	2	**2**		2									3		
3	2	3	3	**2**		2	3							3		2
4	3	3	3	2	1	**2**	3	3	3				3	3	2	
5	3	3		2	3							**3**		3		
6	3	3		2						3		3		**3**		
Avg	**2.84**	**2.67**	**2.67**	**2.0**	**2**	**2**	**3**	**3**	**3**	**3**	**0**	**3**	**3**	**3**	**2**	**2**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L5, CO3 – L3, CO4 – L4, CO5 – L2 and CO6 – L4

Future Course Mapping:

High Speed Networks, Wireless Networks, Mobile Networks, Cryptography and Network Security, Cyber Security, Blockchain, Software Defined Networks.

Job Mapping:

Network Engineer, Network Stack Developers, Application Developer.

Program Elective Courses (PEC1)

Syllabus Template **FF No. : 654**

CS3111::Web Technologies

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**

Lab: 2 Hours/Week

Course Prerequisites: Computer Networks

Course Objectives:

1. To Implement server-side web development using PHP, MySQL, and Laravel MVC framework.
2. To Build interactive React interfaces using components, state, props, routing, and Redux.
3. To Develop Java web applications using Servlets, JSP, MVC architecture, and database integration.
4. To Create backend systems using Spring Boot with REST APIs, security, and deployment.
5. To Apply advanced, React features like TypeScript, hooks, SSR, and performance optimization.
6. To Build scalable Node.js applications using Express, APIs, modules, and databases.

Course Relevance:

The key technology of the information age is global communication. Web technology is a truly global area of study as it enables global communication with the help of web sites. Web technologies are the backbone of all IT infrastructures and their applications in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world. The main objective of the course is present the basic web technology concepts that are required for developing web applications. The key technology components are descriptive languages, server-side program elements and client-side program elements. In addition, the course gives specific contents that are beneficial for developing web-based solutions, like relational data-base communication basics and information security principles and approaches. Most of the jobs available in the IT industries are web technology related.

Section 1: Topics/Contents

Unit-I Server-Side Technologies **(6 Hours)**

PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, File Handling, Exception Handling, E-mail validations, MySQL with PHP, Laravel - The PHP Framework For Web Artisans.

Unit-II React **(6 Hours)**

Introduction to React, react component, JSX, Render function, Component API, Component lifecycle, State, Props, Component composition, pass data from parent to child, Pass data from child to parent, Component styling, Forms, Events, Refs, Keys, Router, Flux, Redux[4 Hrs.]

Unit-III Server-Side Technologies **(7 Hours)**

Servlet: Servlet architecture overview, A "Hello World" servlet, Servlets generating dynamic content, Servlet life cycle, parameter data, sessions, cookies, URL rewriting, other Servlet capabilities, data storage, Servlets concurrency, databases (MySQL) and Java Servlets.

JSP: Introduction to Java Server Pages, JSP and Servlets, running JSP applications, Basic JSP, JavaBeans classes and JSP, Support for the Model-view-controller paradigm, JSP related technologies.

Section2: Topics/Contents

Unit-IV Spring Boot **(8 Hours)**

Spring Framework, Spring Boot Framework, Installing Spring Boot, Build Tool Maven/Gradle/Ant, Core Features, Spring Security, Web Applications, JPA for database connectivity, working with SQL and NoSQL, Messaging, Testing, Deploying Spring Boot Applications, Monitoring and Testing. POSTMAN Tool for API testing.

Unit V Advanced React **(7 Hours)**

TypeScript integration, complex performance tuning, server-side rendering, and robust state libraries. Core Advanced Concepts & Hooks. State Management and API Integration. Advanced Component Patterns and Architecture

Unit-VI Node JS **(8 Hours)**

Introduction to Node JS, Installation of Node JS, Node JS Modules, Node Package Manager (NPM), Creating Web server, File System, Express JS, Serving Static Resources, Database connectivity.

List of Practical's (Minimum Six):

1. Design and develop a responsive website with the help of any code generation AI tool like ChatGPT to calculate Electricity bill using PHP. Condition for first 50 units – Rs. 3.50/unit, for next 100 units – Rs. 4.00/unit, for next 100 units – Rs. 5.20/unit and for units above 250 – Rs. 6.50/unit. You can make the use of bootstrap as well as jQuery.

2. Build an advanced real-time clock dashboard application using React that combines Analog Clock, Digital Clock, Time zone Management, Alarm Features, and Dynamic UI Controls into a single responsive system. The application should simulate a professional world-clock dashboard

capable of displaying live time updates across multiple regions while maintaining smooth animations and optimized performance.

3. Design and develop a responsive website to calculate Electricity bill with the help of any code generation AI tool like ChatGPT with Servlet and JSP. Condition for first 50 units – Rs. 3.50/unit, for next 100 units – Rs. 4.00/unit, for next 100 units – Rs. 5.20/unit and for units above 250 – Rs. 6.50/unit. You can make the use of bootstrap as well as jQuery.

4. Design and develop a responsive website to prepare one semester result of VIT students using REACT with the help of with the help of any code generation AI tool like ChatGPT, Take any four subjects with MSE Marks (30%) ESE Marks (70%).

5. Design and develop a responsive website to prepare one semester result of VIT students using JavaScript, Spring-Boot MySQL/MongoDB/Oracle. Take any four subjects with MSE Marks (30%) ESE Marks (70%).

6. With the help of ChatGPT design and develop a responsive website for an online book store using Node JS and MySQL/ MongoDB/Oracle having 1) Home Page2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

7. Design and develop a responsive website for an online book store using REACT, Spring boot and MySQL/ MongoDB/Oracle having 1) Home Page2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

8. Build an interactive web-based UML Class Diagram Generator using React that enables users to visually create UML class diagrams and automatically generate corresponding Java source code based on the designed architecture.

9. Build a scalable and real-time cricket score management system using Spring Boot that fetches, processes, and displays live cricket match information through REST APIs and a responsive frontend dashboard. The application should simulate a live cricket scoring platform capable of handling ongoing matches, score updates, player statistics, and match summaries efficiently.

List of Course Project areas:

1. Develop a responsive web application for Student Grievance System
2. Develop a responsive web application for Workflow Management System for MNC
3. Develop a responsive Gaming Website
4. Develop a responsive web application to help farmers to solve their farming problems
5. Develop a responsive web application for GST Billing Software for Small Business

6. Develop a responsive web application for online Crime Reporting System using PHP
7. Develop a responsive web application for online College Voting System
8. Develop a responsive web application for online Loan Processing System for Farmers.
9. Develop a responsive web application for restaurant food order management
10. Develop a responsive web application for e-book shop
11. Develop a responsive web application for on-line music store
12. Develop a responsive web application for guest visiting management to your society
13. Develop a responsive web application for web search engine

Text Books: ***(As per IEEE format)***

1. *Kumar, A., Web technologies, CRC press, 2019*
2. *Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019*
3. *Martin, M.G., Programming for Beginners: 6 Books in 1 – Swift+PHP+Java+Javascript+Html+CSS: Basic Fundamental Guide for Beginners, independently published, 2018*
4. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, O'Reilly Media; 5th edition, 2018*
5. *Kohli, S., Web Technologies, PPB Publications, 2015*
6. *Adam Bretz& Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015*
7. *GiulioZambon, "Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013*
8. *Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, Fifth Edition, 2015*
9. *Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035.*
10. *Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2008*

Reference Books: ***(As per IEEE format)***

1. *Marty Hall, Larry Brown,"Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978- 0130897930.*
2. *H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.*
3. *Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006.*
4. *Xue Bai et al: The web Warrior Guide to Web Programming, Thomson, 2003*

MOOCs Links and additional reading material:

1. www.w3.org
2. HTML, The Complete Reference
3. www.htmlref.com
4. w3schools.org
5. php.net/ https://jquery.com/
6. developer.mozilla.org/en-US/docs/AJAX
7. www.tutorialspoint.com/css/
8. PHP: Data Structures - Manual -----
9. docs.spring.io/spring-boot/docs/current/reference/html/
10. nodejs.org/en
11. react.dev

Course Outcomes:

After completing this course students will able to:

1. Build dynamic web applications using PHP and Laravel with server-side scripting and database integration.
2. Write responsive and interactive user interfaces using React components, state, props, and routing concepts.
3. Create robust Java web applications using Servlets and JSP with MVC architecture and session management.
4. Develop secure and scalable backend applications using Spring Boot with REST APIs and database connectivity.
5. Design and develop advanced React applications using TypeScript, hooks, performance optimization, and state management libraries.
6. Design, develop and formulate efficient server-side applications using Node.js and Express with APIs, modules, and database integration.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS3215.1	2	2	2									3	1		2	
CS3215.2	2	2	2									3	1		2	
CS3215.3	3	3	3	1	2	2	2	2				3	1		3	2
CS3215.4	3	3	3	1	2	2	2	2				3	1		3	2
CS3215.5	3	3	3	1	2	2	2	2			2	3	1		3	2
CS3215.6	3	3	3	1	2	2	2	2	2	3	2	3	1		3	2
Average	**2.66**	**2.66**	**2.66**	**1.0**	**2**	**2**	**2**	**2**	**2.0**	**3.0**	**2.0**	**3.0**	**1.0**		**2.66**	**2.0**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Cloud Computing, Distributed System, Mobile Application Development

Job Mapping:

Software Engineer, Web Developer, IT Engineer, UI Developer

Syllabus Template **FF No. : 654**

CS3112: Mainframe Technology

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**

Lab: 2 Hours/Week

Course Prerequisites: Problem Solving Skills

Course Objectives:

1. To introduce IBM Z Mainframe Technology.
2. To learn industry-critical enterprise computing skills.
3. To introduce IBM Mainframe Operating System z/OS.
4. To learn z/OS Access and Interaction Basics.
5. To study the Operating System Environment for computing.
6. To learn z/OS REXX, COBOL Programming

Course Outcomes:

Students will be able to:

1. Understanding of IBM Z Mainframe Technology.
2. Apply the industry-critical enterprise computing skills.
3. Apply IBM Mainframe Operating System z/OS for Large Enterprise.
4. Apply z/Os Access and Interaction Basics.
5. Design and develop a program for Job Control Language using z/OS.
6. Design and develop a program for REXX, COBOL using z/OS.

Course Relevance:

A mainframe technology course remains highly relevant for building a stable, high-paying IT career. Enterprises in banking, insurance, and government rely on mainframe systems for mission-critical operations, massive transaction volumes, and high-security needs, creating steady demand for skilled professionals to maintain and modernize these systems.

Syllabus

Theory

Unit 1: Introduction to IBM Z Mainframe Technology (4 Hours)

Introducing IBM Z Hardware, IBM z16, IBM z16 single frame, IBM z16 rack mount, IBM z15 T01 and T02 machines, Applications of IBM Z, Modern Mainframe innovations: z15 and z16 systems, Containerization and Kubernetes on Mainframes, Mainframe Hardware components such as Central Processing Unit (CPU), Memory and Storage Systems, Input/Output Devices, Case studies of Mainframe implementations.

Unit 2: Learn Industry-Critical Enterprise Computing Skills (4 Hours)

IBM Z Mainframe Skills Depot: Mainframe System Administrator, Mainframe Application Developer/Programmer, Mainframe Modernization Architect, Database Administrator (DBA), Storage Administrator, System Programmer (SysProg), Network Administrator, Mainframe DevOps Engineer, Cybersecurity Engineers, to name a few job roles in the vast and amazing world of the IBM Z Mainframes.

Unit 3: z/OS Introduction (6 Hours)

z/OS: Essential Technology of Large Enterprises, Introducing IBM Mainframe Operating Systems, z/OS - IBM Z Flagship Operating System, z/OS Professional Documentation, Architecture of IBM Z systems, networking, and security aspects.

Unit 4: z/OS Access and Interaction Basics (7 Hours)

Time Sharing Option, Interactive System Productivity Facility (ISPF) Editor, System Display and Search Facility (SDSF), TN3270, SSH, FTP. Introduction to Structured Programming
Datasets and Data Access Methods: Dataset Names and Types, Sequential Dataset, Partitioned Datasets, VSAM - Virtual Storage Access Method, Disk Storage, Volume Labels, VTOC, Extents, System Utilities and Tools, Dataset management utilities (IEBCOPY, IEFBR14, IEBGENER, IDCAMS, etc.), DFSORT basic and examples

Unit 5: Operating System Environment (7 Hours)

Unix System Services (USS), JCL - Job Control Language, JES - Job Entry Subsystem, System Catalog, Components, Messages, System Log, Virtual Storage and Address Spaces, System and User Address Space Management, Key Controlled Protection for Address Spaces, Security, RACF, Communications Server, z/OS Operator Commands, z/OS Initial Load Program Process, Virtualization and Cloud computing.
Data Definition (DD) Statement: Connecting a physical dataset to a program's internal logical filename. SPACE allocation (Primary, Secondary, Directory blocks for PDS, and RLSE).

Unit 6: Mainframe Programming (8 Hours)

z/OS REXX programming, Learning COBOL programming with VS code, z/OS Unix System Services (USS), JCL (Job Control Language), Interactive System Productivity Facility (ISPF), Java on z/OS, Virtual Storage Access Method (VSAM), Ansible Automation, creating fully automated CI/CD pipelines involving components residing on z/OS, Zowe, Big Data and Analytics: Hadoop on Mainframes, Machine Learning Integrations. Mainframe Application Lifecycle Management. Agile project Methodology.

Level Numbers: Structural levels (01-49), elementary items (77), condition names (88). Computational data types like Packed Decimal (COMP-3). Looping/Iteration: PERFORM, UNTIL, TIMES, USING (Parameters), Perform <Para Name1> Through <Para Name2>, REDEFINES and RENAMES clauses, Arrey (single and double Dimension), Static and Dynamic Table in Working Storage.

Syllabus
Laboratory

List of Assignments:

Job Control Language (JCL) Assignments:

1) Dataset operations (Create, Delete, Rename, Copy, Catalog, Uncatalog, etc.)

2) Submit Job, Redirect Output, Check Job Status and Output.

3) Condition code handling, JCL overrides.

4) Creating an In-stream PROC with symbolic parameters, calling it multiple times based on return codes.

COBOL Programming Assignments:

1) Control Structures: If-Else Condition and Simple Loop (PERFORM UNTIL)

2) File Handling: Sequential and Indexed (VSAM)

3) Perform varying, Error handling, etc.

4) Create a file using Data definitions using precise byte alignments for personal and business customers / accounts.

Syllabus
Course Project

Sample Project List: (https://openmainframeproject.org/)

1) To Write a COBOL application program that reads input from three files, merges their contents, and writes to an output file containing one sorted list.

2) Demonstrate file-handling capabilities using a REXX program.

3) Write a Cobol Application program to read files (ATM, POS and Dated Transactions) to update Account file.

CO-PO Mapping

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1	2							2	2	3		
CO2	3	3	2	1	3				1			2	3	3	2	
CO3	3	3	2	2	3							2	3	3	2	1
CO4	2	3	2	2	3				1	1		2	3	3	2	1
CO5	3	3	3	2	3				1	1	1	2	3	3	3	2
CO6	3	3	3	2	3				2	2	1	3	3	3	3	2
Average	2.83	2.83	2.17	1.67	2.83				1.25	1.33	1.00	2.17	2.83	3.00	2.40	1.50

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L3, CO2– L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Courses Mapping:

Advanced Mainframe Technologies.

Job Mapping:

Mainframe Technology is widely used for large scale enterprises such as Banking, Insurance Air Traffic Control and various large- and small-scale enterprises. It is the backbone for large scale enterprises which needs high security and zero down time.

For MOOCs and other learning Resources

Unit-1:

- **https://www.ibm.com/z/education**
- **https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html**

Unit-2:

- **https://www.youtube.com/watch?v=aEXHOBYv56E**

Unit-3:

- **https://www.youtube.com/watch?v=aEXHOBYv56E**
- **https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html**

Unit-4:

- **https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html**

Unit-5:

- **https://learn.ibm.com/course/view.php?id=9890**
- **https://publibz.boulder.ibm.com/zoslib/books/tutorials/introjcl/index.html**

Unit-6:

- **https://openmainframeproject.org/projects/cobol-programming-course/**
- **https://learn.ibm.com/course/view.php?id=9890**
- **https://learn.ibm.com/course/view.php?id=7552**
- **https://learn.ibm.com/course/view.php?id=8722**
- **https://publibz.boulder.ibm.com/zoslib/books/tutorials/introjcl/index.html**

E-Resources:

- **https://ibmzxplore.influitive.com/users/sign_in**
- **https://www.ibm.com/z/education**
- **https://www.tutorialspoint.com/cobol/cobol_overview.htm**

MOOC Courses:

1. **https://www.ibm.com/training/**
2. **https://learn.ibm.com/**
3. **https://www.ibm.com/z/education**
4. **https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html**

Syllabus Template **FF No. : 654**

CS3113: Basics of Game Development

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**

Lab: 2 Hours/Week

Course Prerequisites: Knowledge of OOP, Basic Java Script,

Course Objectives: Student will

1. learn Game Mechanics and web based game development concepts.
2. learn C# OOP fundamentals in order to do Unity 3D scripting.
3. learn programming patterns that are generally used in game development.
4. learn the basics of game services.
5. learn the basics of collision physics in gaming.

Course Relevance:

This course introduces students to the fundamentals of modern game development by combining theoretical concepts with practical implementation on both web-based and standalone game development platforms. It provides exposure to the gaming ecosystem, game design principles, programming patterns, game physics, networking services, and asynchronous programming techniques that are widely used in the game industry. Through hands-on development of games and analysis of real-world gaming scenarios, students gain the knowledge and skills required to design, develop, test, and deploy interactive games while adopting industry-standard tools, frameworks, and software engineering practices.

Theory

Unit 1: Game Domain

Gaming as business & Global market, Introduction to Gaming, Gaming Ecosystem and opportunities, Player time spending, what kind of games succeed, monetizing game, how to target the gaming crowd, what is working for real money gaming, game testing, types of game testing, Realism and compliance testing, multiplayer and network testing

Unit 2: Web based Game programming

Basics of HTML, CSS, Java Script, introduction to phaser.io js library

Unit 3: Standalone Game Platform

C# fundamentals with all OOPs concepts, Introduction to Unity 3D Engine, C# scripting in Unity 3D Engine

Unit 4: Programming Design Patterns in Game Development

Observer Design Pattern, State Design Pattern, Command Design Pattern, Decorator Design Pattern

Unit 5: Game Services

Choosing a Gamer Service, Basic Setup, Lobbies and Match making, Networking, Player Statistics, Player Achievements Leaderboards, Other Service,

Unit 6: Game Physics: Collision Detection

The Collision Detection Pipeline, Broad Phase Collision Detection, Requirements, Bounding Volume Hierarchies, Hierarchies Building the Hierarchy, Sub-object Hierarchy, Spatial Partitioning, Binary Space Partitioning, Oct-Trees and Quad-Trees, Grids, Multiresolution Maps

Laboratory

List of Experiments

1.Snake Game

Develop a simple Snake game using JavaScript and H5 and a game framework such as Phaser.io. The game should include the following features:

1. A snake that can move in four directions (up, down, left, right) using the arrow keys.
2. Food items that randomly appear on the game field.
3. The snake grows longer each time it eats a food item.
4. The game ends when the snake collides with itself or the walls.
5. A scoring system that tracks the number of food items eaten.

2.Tetris Game

Develop a simple Tetris game using C# and a game framework such as Unity. The game should include the following features:

1. Randomly generated Tetriminos (tetrominoes).
2. The ability to move and rotate Tetriminos.
3. The game field (grid) should be able to detect filled rows and clear them.
4. The game should end when the Tetriminos reach the top of the grid.
5. A scoring system that tracks the number of cleared lines.

3. State Machines in Gaming

Implementation Related to State Machines:
Implement a state machine to manage the state of a traffic light system.
Write the code for a traffic light state machine with states like “Green,” “Yellow", "Red” and transitions between these states.

How would you handle asynchronous actions within a state machine?
Provide examples of handling asynchronous actions (like API calls) within a state machine using JavaScript.

4. Scenario Based statement

Describe a scenario where you would use JavaScript for UI and C# for game logic in a game.
Provide an example where this separation of concerns is beneficial and explain how the integration would work.

How would you debug issues in a game with integrated JavaScript and C# components?
Outline the tools and techniques you would use to debug problems that arise from the interaction between JavaScript and

How would you implement cross-platform support for a game using JavaScript and C#?
Describe strategies for ensuring that a game runs smoothly on multiple platforms (e.g., web, mobile, desktop).

5. Core Gaming (Understanding, Analysis and Presentation)

Flowchart/Presentation
Title: Enhancing Player Engagement in Candy Crush/Temple Run/Puzzle game Using Design Patterns

Observer Pattern
Description: Real-time notifications for friend activities and game events.
Impact: Increases social interaction and competitive spirit.
Visual: Diagram showing notification flow from game server to players.

State Pattern
Description: Manage different game states for varied gameplay.
Impact: Keeps the game fresh and challenging.
Visual: Flowchart showing transitions between Normal Play, Special Event, and Timed Challenge states.

Command Pattern
Description: Allow players to undo moves and activate abilities.
Impact: Enhances control and reduces frustration.

Visual: Sequence diagram illustrating the Command pattern for undoing a move.

Decorator Pattern
Description: Add dynamic features like power-ups and skins.
Impact: Provides customization and boosts player excitement.
Visual: Diagram showing how decorators add new features to game components.

Conclusion
Summary: Recap of the suggested patterns and their benefits.
Next Steps: Implementation plan and expected outcomes.

6. Async and sync Implementation in Gaming
How do you define an asynchronous function in JavaScript?
Provide examples of defining asynchronous functions using `async` and `await` keywords.

What are Promises and how are they used in JavaScript?
Explain the concept of Promises and their role in managing asynchronous operations.

Implement a basic game loop using synchronous functions in JavaScript.
Write code for a simple game loop that updates game state and renders frames synchronously.

Convert the above synchronous game loop into an asynchronous one.
Rewrite the game loop using `requestAnimationFrame` and asynchronous functions to prevent blocking the main thread.

How would you handle asynchronous loading of game assets (e.g., images, sounds) in JavaScript?
Provide code examples for loading game assets asynchronously using Promises or async/await.

Implement a function to fetch high scores from a server asynchronously in a JavaScript game
Write a function that uses `fetch` or `XMLHttpRequest` to retrieve high scores from an external server without blocking the game.

Describe how you would manage asynchronous player inputs in a multiplayer game
Explain techniques for handling player inputs asynchronously, ensuring that the game remains responsive and updates in real-time.

Course Outcomes

Upon completion of this course, students will be able to,

1. Explain the fundamentals of the gaming industry, game development lifecycle, gaming platforms, and business models.
2. build simple interactive games using appropriate game development frameworks and programming languages for web and standalone platforms..
3. Apply game programming concepts, software design patterns, and state management techniques to develop maintainable game applications.
4. Apply essential game development features such as physics, collision detection, game services, networking, and asynchronous programming.
5. Analyze game architectures, programming approaches, and implementation strategies for solving real-world game development scenarios and enhancing player experience..
6. Design, develop, and present game-based solutions by integrating appropriate game development tools, programming techniques, and industry practices.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2	3	-	3	1	-	-	-	-	-	-	3	1	-	-
CO2	3	2	3	-	3	1	-	-	2	-	-	-	3	2	-	-
CO3	3	2	3	-	3	1	-	-	2	-	-	-	3	3	2	-
CO4	3	2	3	-	3	2	1	1	1	-	1	-	3	2	-	-
CO5	3	2	3	-	3	2	1	2	2	-	2	2	3	1	1	1
CO6	3	2	3	-	3	3	1	3	3	3	3	3	3	1	-	2
Average	**3**	**2**	**3**	**0**	**3**	**1.67**	**2**	**2**	**2**	**3**	**2**	**2.5**	**3**	**1.67**	**1.5**	**1.5**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium–0.65 L4 – Somewhat difficult–0.6 L5 – Difficult–0.55

CO1 – L1, CO2– L2, CO3 – L3, CO4 – L3, CO5 – L4 and CO6 – L4

Future Courses Mapping:

Advanced Game Development, Computer Graphics, ARVR

Job Mapping:

Game Developer, Game Artist, Game Tester

Books and E-Resources

Text books

1. Colt McAnlis, et al., “HTML5 Game Development Insights”, 2014, Apress, ISBN: 978-1-4302-6698-3. (https://doi.org/10.1007/978-1-4302-6698-3)
2. Joshua Glazer, Sanjay Madhav, “ Multiplayer Game Programming, Architecting Networked Games”, 2016, Addison-Wesley, ISBN-13: 978-013-403430-0
3. Ian Millington, “Game Physics Engine Development”, 2007, Morgan Kaufmann, ISBN-13: 978-0-12-369471-3

For MOOCs and other learning Resources :

1. Udemy
2. Coursera

Program Elective Courses 2 (PEC2)

Course Code	Course Name
MD3141	AWS Cloud Technology Consultant
MD3101	IBM Full Stack Software Developer
MD3102	Meta Back-End Developer
MD3106	Microsoft Power BI Data Analyst
MD3112	Google Cyber Security
MD3113	Google Data Analytics
MD3121	IBM DevOps and Software Engineering

Syllabus Template **FF No. : 654**

MD3141: AWS Cloud Technology Consultant

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 10 Hrs / Week

Course Link: https://www.coursera.org/professional-certificates/aws-cloud-technology-consultant

Course Relevance:

This is a professional certification course designed for Computer Engineering professionals seeking to specialize in cloud-native strategy and architectural consulting. This course has a high relevance in enterprise environments, digital transformation projects, and professional services, serving as a critical bridge between core technical implementation and high-level business logic. As cloud ecosystems become increasingly complex, the ability to translate business drivers into optimized technical blueprints is a vital prerequisite for modern architects. A mastery of these consulting methodologies allows professionals to oversee cloud lifecycles and design multi-layered infrastructures conceptual, logical, and physical ensuring that the resulting platforms are not only functional but also aligned with cost-efficiency, scalability, and long-term organizational goals. In this course students will:

1. Understand and translate business goals and drivers into appropriate technical solutions
2. Provide technical direction and oversight to cloud implementation teams
3. Learn how to gather technical requirements, assess client capabilities and analyze findings to provide appropriate cloud solution recommendations
4. Create the conceptual, logical and physical design for cloud-based solutions for infrastructure and platforms

Syllabus: It has 9 modules:

1. Introduction to Information Technology and AWS Cloud
2. AWS Cloud Technical Essentials
3. Providing Technical Support for AWS Workloads
4. Developing Applications in Python on AWS
5. Skills for Working as an AWS Cloud Consultant
6. DevOps on AWS and Project Management
7. Automation in the AWS Cloud
8. Data Analytics and Databases on AWS
9. Capstone: Following the AWS Well Architected Framework

Syllabus Template **FF No. : 654**

MD3101: IBM Full Stack Software Developer

Credits: 3 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week
Course Link: https://www.coursera.org/professional-certificates/ibm-full-stack-cloud-developer

Course Relevance:

This course has a high relevance in all domains of software engineering such as in IT Industries, startups, and product-based companies as a primary skill-building course. Full Stack development is a crucial part of modern software engineering as it allows developers to manage the entire application lifecycle from UI design to cloud deployment. A thorough understanding of both client-side and server-side technologies, combined with cloud-native tools, enables a developer to build versatile, scalable, and efficient software solutions that meet contemporary industry demands. In this course students will:

1. Develop with front-end development languages and tools such as HTML, CSS, JavaScript, React and Bootstrap
2. Program applications using back-end languages and frameworks like Express, Node.js, Python, Django, etc.
3. Deploy and scale applications using Cloud Native methodologies and tools like Containers, Kubernetes, Microservices and Serverless Functions
4. Build your GitHub portfolio by applying your Full Stack Cloud Development skills with multiple labs and hands-on projects, including a capstone

Syllabus: It has 15 modules:

1. Introduction to Software Engineering
2. Introduction to Cloud Computing
3. Introduction to HTML, CSS, & JavaScript
4. Getting Started with Git and GitHub
5. Developing Front-End Apps with React
6. Developing Back-End Apps with Node.js and Express
7. Python for Data Science, AI & Development
8. Developing AI Applications with Python and Flask
9. Django Application Development with SQL and Databases
10. Introduction to Containers w/ Docker, Kubernetes & OpenShift
11. Application Development using Microservices and Serverless
12. Full Stack Application Development Capstone Project
13. Full Stack Software Developer Assessment
14. Generative AI: Elevate your Software Development Career
15. Software Developer Career Guide and Interview Preparation

Syllabus Template **FF No. : 654**

MD3102: Meta Back-End Developer

Credits: 3 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week
Course Link: https://www.coursera.org/professional-certificates/meta-back-end-developer

Course Relevance:

This is a core professional course for Computer Engineering and backend specialization branches. This course has a high relevance in all domains of software architecture such as in Fintech, E-commerce, and SaaS industries as a vital prerequisite course. Back-end development is a crucial part of software ecosystems as it allows developers to build the "brain" of the application where data processing and security reside. A wise selection of server-side tools and database optimization techniques can improve the performance of a web application in a more scalable and secure way. In this course students will:

1. Gain the technical skills required to become a qualified back-end developer
2. Learn to use programming systems including Python Syntax, Linux commands, Git, SQL, Version Control, Cloud Hosting, APIs, JSON, XML and more
3. Build a portfolio using your new skills and begin interview preparation including tips for what to expect when interviewing for engineering jobs
4. Learn in-demand programming skills and how to confidently use code to solve problems

Syllabus: It has 9 modules:

1. Introduction to Back-End Development
2. Programming in Python
3. Version Control
4. Introduction to Databases for Back-End Development
5. Django Web Framework
6. APIs
7. The Full Stack
8. Back-End Developer Capstone
9. Coding Interview Preparation

Syllabus Template **FF No. : 654**

MD3106: Microsoft Power BI Data Analyst

Credits: 3 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week
Course Link: https://www.coursera.org/professional-certificates/microsoft-power-bi-data-analyst

Course Relevance:

This is a critical professional course for Data Science, Business Management, and Information Technology branches. This course has a high relevance in all domains of corporate operations such as in Finance, Marketing, and Supply Chain Management as a vital prerequisite course. Data visualization and business intelligence are crucial parts of modern decision-making as they allow organizations to convert raw data into strategic assets. A wise selection of visualization techniques and data modeling strategies can improve the clarity of business reporting and organizational efficiency in a more impactful way. In this course students will:

1. Learn to use Power BI to connect to data sources and transform them into meaningful insights.
2. Prepare Excel data for analysis in Power BI using the most common formulas and functions in a worksheet.
3. Learn to use the visualization and report capabilities of Power BI to create compelling reports and dashboards.
4. Demonstrate new skills with a capstone project and prepare for the industry-recognized Microsoft PL-300 Certification exam.

Syllabus: It has 9 modules:

1. Preparing Data for Analysis with Microsoft Excel
2. Harnessing the Power of Data with Power BI
3. Extract, Transform and Load Data in Power BI
4. Data Modeling in Power BI
5. Data Analysis and Visualization with Power BI
6. Creative Designing in Power BI
7. Deploy and Maintain Power BI Assets and Capstone project
8. Microsoft PL-300 Exam Preparation and Practice

Syllabus Template **FF No. : 654**

MD3112: Google Cyber Security

Credits: 3 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week
Course Link: https://www.coursera.org/professional-certificates/google-cybersecurity

Course Relevance:

This is a foundational professional course for Computer Engineering, Information Technology, and Network Security branches. This course has a high relevance in all domains of modern technology such as in Banking, Government, and E-commerce as a critical prerequisite course. Cybersecurity is a crucial part of the digital landscape as it allows organizations to maintain data privacy and system integrity against evolving global threats. A wise selection of security controls and mitigation tools can improve the safety of a computer network or organizational database in a more resilient way. In this course students will:

1. Understand the importance of cybersecurity practices and their impact for organizations.
2. Identify common risks, threats, and vulnerabilities, as well as techniques to mitigate them.
3. Protect networks, devices, people, and data from unauthorized access and cyberattacks using Security Information and Event Management (SIEM) tools.
4. Gain hands-on experience with Python, Linux, and SQL.

Syllabus: It has 9 modules:

1. Foundations of Cybersecurity
2. Play It Safe: Manage Security Risks
3. Connect and Protect: Networks and Network Security
4. Tools of the Trade: Linux and SQL
5. Assets, Threats, and Vulnerabilities
6. Sound the Alarm: Detection and Response
7. Automate Cybersecurity Tasks with Python
8. Put It to Work: Prepare for Cybersecurity Jobs
9. Accelerate Your Job Search with AI

Syllabus Template **FF No. : 654**

MD3113: Google Data Analytics

Credits: 3 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week
Course Link: https://www.coursera.org/professional-certificates/google-data-analytics

Course Relevance:

This is a comprehensive professional course for Computer Engineering, Data Science, and Business Intelligence branches. This course has a high relevance in all domains of modern industry such as in Marketing, Finance, and Operations as a fundamental prerequisite course. Data Analytics is a crucial part of modern business as it allows organizations to transform raw information into actionable intelligence. A wise selection of analytical tools and visualization techniques can improve the performance of a business strategy or operational workflow in a more evidence-based and impactful way. In this course students will:

1. Gain an immersive understanding of the practices and processes used by a junior or associate data analyst in their day-to-day job
2. Learn key analytical skills (data cleaning, analysis, & visualization) and tools (spreadsheets, SQL, Python, Tableau)
3. Understand how to clean and organize data for analysis, and complete analysis and calculations using spreadsheets, SQL and Python
4. Learn how to visualize and present data findings in dashboards, presentations and commonly used visualization platforms

.

Syllabus: It has 9 modules:

1. Foundations: Data, Data, Everywhere
2. Ask Questions to Make Data-Driven Decisions
3. Prepare Data for Exploration
4. Process Data from Dirty to Clean
5. Analyze Data to Answer Questions
6. Share Data Through the Art of Visualization
7. Introduction to Data Analysis Using Python
8. Google Data Analytics Capstone: Complete a Case Study
9. Accelerate Your Job Search with AI

Syllabus Template **FF No. : 654**

MD3121: IBM DevOps and Software Engineering

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 10 Hrs / Week

Course Link: https://www.coursera.org/professional-certificates/devops-and-software-engineering

Course Relevance:

This is a high-impact professional course for Computer Engineering, Software Engineering, and Systems Administration branches. This course has a high relevance in all domains of modern technology such as in Cloud Service Providers, SaaS Enterprises, and Fintech as a critical prerequisite course. DevOps is a crucial part of the software engineering landscape as it allows teams to bridge the gap between development and operations, ensuring rapid, reliable software releases. A wise selection of automation tools and containerization strategies can improve the deployment frequency and system stability of a software product in a more agile and scalable way. In this course students will:

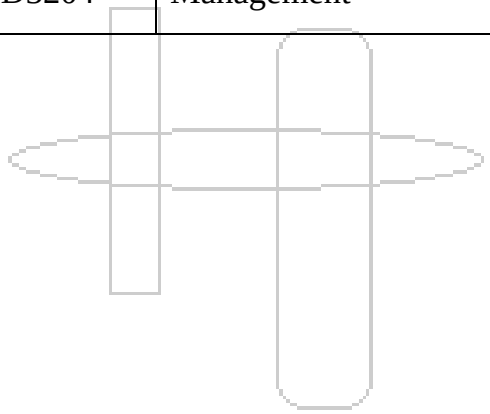
1. Develop a DevOps mindset, practice Agile philosophy & Scrum methodology - essential to succeed in the era of Cloud Native Software Engineering
2. Create applications using Python language, using various programming constructs and logic, including functions, REST APIs, and libraries
3. Build applications composed of microservices and deploy using containers (e.g. Docker, Kubernetes, and OpenShift) & serverless technologies
4. Employ tools for automation, continuous integration (CI) and continuous deployment (CD) including Chef, Puppet, GitHub Actions, Tekton and Travis.

Syllabus: It has 15 modules:

1. Introduction to DevOps
2. Introduction to Cloud Computing
3. Introduction to Agile Development and Scrum
4. Introduction to Software Engineering
5. Getting Started with Git and GitHub
6. Hands-on Introduction to Linux Commands and Shell Scripting
7. Python for Data Science, AI & Development
8. Developing AI Applications with Python and Flask
9. Introduction to Containers w/ Docker, Kubernetes & OpenShift
10. Application Development using Microservices and Serverless
11. Introduction to Test and Behavior Driven Development
12. Continuous Integration and Continuous Delivery (CI/CD)
13. Application Security for Developers and DevOps Professionals
14. Monitoring and Observability for Development and DevOps
15. DevOps Capstone Project

Open Elective Courses (OE)

Course Code	Course Name
MD3201	Fintech
MD3202	Digital marketing
MD3203	Entrepreneurship
MD3204	Management



Syllabus Template **FF No. : 654**

MD3201: FinTech

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**

Course Relevance:
The subject is highly relevant for engineering students as it provides knowledge of how technology is transforming the financial sector through digital banking, payments, data analytics, artificial intelligence, blockchain, and automation. It helps students understand financial systems, market operations, risk management, and the role of innovation in improving financial services. The subject also introduces important legal, regulatory, and cybersecurity considerations that are essential for developing secure and compliant financial solutions. By combining engineering, finance, and business concepts, FinTech equips students with practical skills that support careers in software development, financial technology, digital banking, cybersecurity, data analytics, and entrepreneurship. This interdisciplinary knowledge prepares graduates to contribute effectively to the growing digital financial ecosystem. In this course students will:

1. **Understand** the role of emerging technologies such as AI, blockchain, and digital payments in transforming financial services.
2. **Analyze** financial markets, instruments, and risk management concepts relevant to fintech applications.
3. **Apply** engineering and data-driven approaches to design and evaluate innovative financial technology solutions.
4. **Assess** regulatory, legal, security, and compliance requirements associated with digital financial systems.
5. **Develop** problem-solving and interdisciplinary skills for careers in fintech, digital banking, financial analytics, and technology-driven financial services.

Syllabus: This subject contains following courses

FinTech: Finance Industry Transformation and Regulation Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/fintech?source=search
Digital Transformation in Financial Services Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/digital-transformation-financial-services?source=search
Financial Markets Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/financial-markets?source=search
FinTech Law and Policy	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/learn/fintechlawandpolicy?source=search

Decentralized Finance (DeFi): The Future of Finance Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/decentralized-finance-duke?source=search
FinTech: Finance Industry Transformation and Regulation Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/fintech?source=search

Syllabus Template **FF No. : 654**

MD3202: Digital Marketing

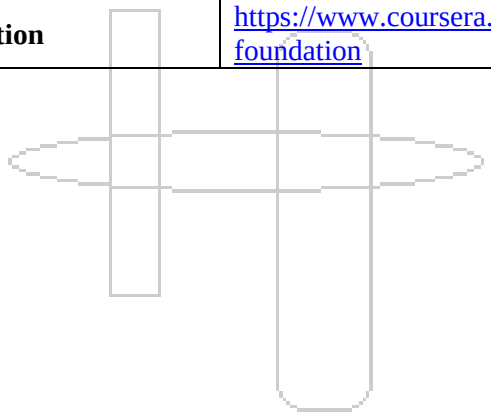
Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

Digital Marketing provides knowledge and skills required to promote products, services, and innovations in the digital era. The subject introduces students to key concepts such as search engine optimization (SEO), social media marketing, content marketing, email marketing, web analytics, and online advertising. Engineering graduates often work on technology-based products and solutions, and understanding digital marketing helps them effectively communicate value to customers and stakeholders. The course also develops analytical skills through the use of data-driven marketing strategies, consumer behavior analysis, and performance measurement tools. Furthermore, it enhances students' understanding of emerging technologies, digital platforms, and business models that drive modern organizations. By combining technical expertise with marketing knowledge, students become better equipped for careers in technology management, entrepreneurship, product development, e-commerce, and digital business. The subject encourages creativity, strategic thinking, and customer-centric problem-solving, enabling engineers to contribute effectively to the growth and competitiveness of organizations in a digitally connected world. In this course students will:

1. Understand the fundamental concepts, tools, and techniques used in digital marketing and online promotion.
2. Apply digital marketing strategies including SEO, social media marketing, content marketing, and email marketing to real-world scenarios.
3. Analyze consumer behavior and marketing campaign performance using web analytics and data-driven decision-making techniques.
4. Develop effective digital marketing plans to enhance brand visibility, customer engagement, and business growth.
5. Evaluate emerging digital technologies and platforms to support innovation, entrepreneurship, and technology-based business solutions.

Syllabus: This subject contains following courses

Foundations of Digital Marketing and E-commerce	https://www.coursera.org/learn/foundations-of-digital-marketing-and-e-commerce
Adobe Digital Marketing	https://www.coursera.org/learn/adobe-digital-marketing
Digital Marketing Operations & Analytics Specialization	https://www.coursera.org/specializations/digital-marketing-operations-analytics
Social Media and Digital Marketing Fundamentals	https://www.coursera.org/learn/social-media-digital-marketing-fundamentals
Introduction to Digital Marketing	https://www.coursera.org/learn/introduction-digital-marketing
Digital Marketing Trends	https://www.coursera.org/learn/digital-marketing-trends
Assess for Success: Marketing Analytics and Measurement	https://www.coursera.org/learn/assess-for-success
Concepts, Strategies, and Analytics in Performance Marketing and Digital Advertising Specialization	https://www.coursera.org/specializations/digital-advertising-strategy
Integrated Marketing Communications: Advertising, Public Relations, Digital Marketing and more	https://www.coursera.org/learn/integrated-marketing-communications
Marketing in a Digital World	https://www.coursera.org/learn/marketing-digital
Marketing Analytics Foundation	https://www.coursera.org/learn/marketing-analytics-foundation



Syllabus Template **FF No. : 654**

MD3203: Entrepreneurship

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

Entrepreneurship is a highly valuable subject for engineering students as it develops the mindset and skills required to transform innovative ideas into successful business ventures. The subject introduces students to key concepts such as opportunity identification, business model development, innovation management, startup creation, market analysis, financial planning, and leadership. Engineering students often possess strong technical knowledge, and entrepreneurship helps them understand how to commercialize technologies and create solutions that address real-world problems. The course encourages creativity, critical thinking, risk assessment, decision-making, and problem-solving abilities that are essential for launching and managing new ventures. It also provides insights into funding strategies, customer needs, product development, and business growth. By integrating technical expertise with entrepreneurial skills, students become better prepared for careers as startup founders, innovators, project leaders, and business professionals. The subject fosters self-reliance, adaptability, and a proactive approach to identifying opportunities in a competitive global market, enabling engineering graduates to contribute to economic development and technological advancement. In this course students will:

1. Identify entrepreneurial opportunities and evaluate the feasibility of innovative business ideas.
2. Develop business models and startup strategies for technology-based products and services.
3. Apply problem-solving, leadership, and decision-making skills in entrepreneurial and organizational contexts.
4. Analyze market needs, customer requirements, and financial aspects for successful venture creation and growth.
5. Demonstrate an entrepreneurial mindset by managing risks, fostering innovation, and pursuing sustainable business opportunities.

Syllabus: This subject contains following courses

Essentials of Entrepreneurship: Thinking & Action	https://www.coursera.org/learn/entrepreneurial-thinking
Technology & Entrepreneurship Specialization	https://www.coursera.org/specializations/technology-and-entrepreneurship
Entrepreneurship: Growing Your Business Specialization	https://www.coursera.org/specializations/entrepreneurship-growing-your-business
Skills for a Modern Entrepreneur or Business Owner Specialization	https://www.coursera.org/specializations/packt-skills-for-a-modern-entrepreneur-or-business-owner
Innovation & Entrepreneurship - From Design Thinking to Funding	https://www.coursera.org/learn/design-thinking-entrepreneurship
Innovation and entrepreneurship Specialization	https://www.coursera.org/specializations/innovation-entrepreneurship
The Entrepreneurial Toolkit Specialization	https://www.coursera.org/specializations/the-entrepreneurial-toolkit

Syllabus Template **FF No. : 654**

MD3204: Management

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

Management subject provides the knowledge and skills required to effectively plan, organize, lead, and control resources in professional and organizational settings. While engineering education develops strong technical expertise, management studies help students understand how to manage projects, teams, operations, and business processes efficiently. The subject introduces key concepts such as strategic planning, organizational behavior, leadership, decision-making, human resource management, quality management, and project management. It enables students to develop communication, teamwork, problem-solving, and analytical skills that are essential for successful careers in industry and business. Management knowledge also helps engineers understand organizational goals, customer requirements, resource utilization, and productivity improvement. Furthermore, it prepares students to take on leadership roles, manage multidisciplinary teams, and contribute to organizational growth and innovation. By combining technical competence with managerial capabilities, engineering graduates become better equipped to handle complex challenges, improve operational performance, and make informed decisions in dynamic and competitive environments. In this course students will:

1. Understand fundamental management principles, organizational structures, and business functions relevant to engineering practice.
2. Apply planning, organizing, leadership, and decision-making skills to manage projects and teams effectively.
3. Analyze organizational challenges and develop strategies for improving productivity, quality, and operational efficiency.
4. Demonstrate effective communication, teamwork, and leadership skills in multidisciplinary professional environments.
5. Evaluate managerial and strategic decisions considering organizational goals, resource utilization, and sustainable growth.

Syllabus: This subject contains following courses

Digital Product Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/uva-darden-digital-product-management?source=search
Project Management: Tools, Approaches, Behavioural Skills Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/project-management-tools-approaches-skills?source=search
Foundations of Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/foundations-management?source=search
Supply Chain Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/supply-chain-management?source=search

Multi-Disciplinary Minor Courses (MDM)

Course Code	Course Name
MDM***	Blockchain Technologies
MDM***	Ethical Hacking
MDM***	Augmented and Virtual Reality
MDM***	Advanced Computer Architecture

Syllabus Template **FF No. : 654**

MDM***: Blockchain Technologies

Credits: 3 **Teaching Scheme: Theory: 2 Hours / Week**
Tutorial: 1 Hours / Week

Prerequisites:

Programming Fundamentals, Data Structures, Operating Systems, Computer Networks, Basic Cryptography

Course Objectives

1. Understand the core principles and architecture of blockchain technology.
2. Explore cryptographic foundations and consensus algorithms that secure blockchain networks.
3. Gain practical knowledge of major blockchain platforms including Bitcoin, Ethereum, and Hyperledger Fabric.
4. Develop and deploy smart contracts and decentralized applications (DApps).
5. Analyze real-world blockchain applications and emerging trends.
6. Evaluate security, scalability, and regulatory challenges in blockchain ecosystems.

Theory

Unit 1: Blockchain Fundamentals and Cryptography (5 Hours)

Introduction to Blockchain: History, Concepts, Types (Public, Private, Permissioned), Blockchain Architecture: Blocks, Chains, Transactions, Ledgers, Cryptographic Foundations: Hash Functions (SHA-256), Digital Signatures (ECDSA), Public/Private Keys, Distributed Systems Basics: Consensus Problem, Byzantine Generals Problem, Game Theory (Nash Equilibrium)

Unit 2: Consensus Mechanisms and Security (5 Hours)

Consensus Algorithms: Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Byzantine Fault Tolerance (BFT), Mining and Incentive Structures, Security Aspects: Double Spending, 51% Attacks, Sybil Attacks, Privacy Techniques: Zero-Knowledge Proofs, Anonymity in Blockchain (Monero, Zcash)

Unit 3: Blockchain Platforms and Protocols (6 Hours)

Bitcoin Protocol: Transactions, Wallets, Mining, Scripting, Ethereum: Ethereum Virtual Machine (EVM), Smart Contracts, Solidity Basics, Enterprise Blockchains: Hyperledger Fabric Architecture, Chaincode, Other Platforms Overview: Ripple, Multichain, Stellar

Unit 4: Smart Contracts and Decentralized Applications (DApps) (4 Hours)

Smart Contract Development: Solidity Language, Contract Lifecycle, Security Best Practices, Development Tools: Remix IDE, Truffle, Hardhat, DApp Architecture and Development: Web3.js, Ethers.js, Metamask Integration, Deployment and Testing of Smart Contracts

Unit 5: Blockchain Use Cases and Industry Applications (4 Hours)

Use Cases: Supply Chain Management, Finance, Healthcare, Voting, Identity Management, Blockchain Integration with IoT, AI, and Cybersecurity, Case Studies: Food Industry Supply Chain, Land Records, Digital Identity

Unit 6: Emerging Trends, Challenges, and Future Directions *(4 Hours)*

Scalability and Interoperability Challenges, Layer 2 Solutions and Sidechains, Regulatory, Legal, and Ethical Considerations, Future Trends: NFTs, DeFi, Web3, Cross-chain Technologies

List of Tutorials-

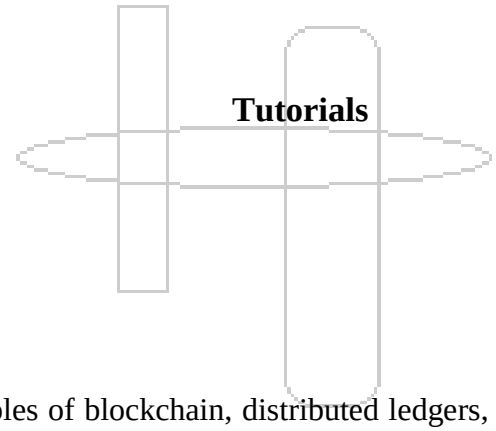
1. Compare centralized vs decentralized systems. Discuss blockchain architecture with real-world examples.
2. Solve numerical/examples on SHA-256 hashing, digital signatures, public/private keys.
3. Trace the lifecycle of a blockchain transaction and explain Merkle Trees.
4. Analyse attacks such as 51% attack, Sybil attack, Double Spending, and discuss mitigation techniques.
5. Compare Bitcoin and Ethereum architecture, transaction flow, mining, gas fees, and scripting.
6. Study Solidity syntax, identify errors in sample contracts, and discuss contract lifecycle.
7. Explain the interaction between Smart Contracts, Web3.js, MetaMask, and Ethereum network.
8. Discuss blockchain applications in healthcare, finance, supply chain, voting, and land records.

List of Lab Experiments

1. **Setting up Bitcoin and Ethereum Wallets**
   Create and configure wallets (e.g., MetaMask, Trust Wallet), perform test transactions to simulate real cryptocurrency transfers as done on exchanges like Binance or Coinbase.
2. **Exploring Blockchain Explorers and Transaction Tracing**
   Use tools like Etherscan and Blockchain.com to trace Bitcoin and Ethereum transactions, similar to how blockchain analysts investigate transaction flows for compliance or fraud detection.
3. **Writing and Deploying Smart Contracts using Solidity on Remix IDE**
   Develop and deploy a smart contract simulating a real-world escrow agreement or crowdfunding contract on Ethereum testnet.
4. **Building a Decentralized Application (DApp) with Web3.js and Metamask**
   Create a simple web-based voting system where users interact with a blockchain through Metamask, similar to how users vote in DAOs (Decentralized Autonomous Organizations).
5. **Implementing Consensus Simulation: Proof of Work (PoW) and Proof of Stake (PoS)**
   Simulate how consensus works by building small-scale PoW (Bitcoin-like) and PoS (Ethereum 2.0-like) systems demonstrating block creation and validation.
6. **Hyperledger Fabric Network Setup and Chaincode Deployment**
   Set up a permissioned blockchain to simulate a food supply chain network for tracking produce, replicating real-world Hyperledger Fabric use by Walmart or IBM Food Trust.
7. **Developing and Deploying a Token (ERC-20 or ERC-721) on Ethereum Testnet**
   Create and deploy a cryptocurrency token (ERC-20) or NFT (ERC-721), similar to tokens used in loyalty programs or NFT art platforms like OpenSea.
8. **Security Testing of Smart Contracts with Common Vulnerabilities**
   Use tools like Mythril or Slither to identify security loopholes (e.g., reentrancy attacks) that have caused real-world incidents such as the infamous DAO hack.
9. **Building a Simple NFT Marketplace Prototype**
   Simulate listing, buying, and transferring NFTs on a mini-marketplace, similar to platforms like OpenSea or Rarible.
10. **Blockchain Integration with IoT: Real-Time Data Logging**
    Simulate an IoT sensor (e.g., environmental temperature) sending data to a blockchain smart contract for immutable record-keeping, similar to real-world applications by IBM or VeChain.
11. **Performance and Gas Cost Analysis of Blockchain Applications**
    Analyze transaction fees and optimize smart contract code to reduce gas consumption, similar to how developers minimize costs for DeFi or NFT platforms.
12. **Interacting with a Layer 2 Solution (e.g., Polygon) for Scalable Blockchain Applications**
    Deploy smart contracts and interact with them on Polygon's testnet to experience reduced transaction fees and increased scalability, as done in popular blockchain games and dApps.

List of Course Projects

1. Decentralized voting system using smart contracts
2. Supply chain tracking on Hyperledger Fabric
3. Land record management with blockchain
4. IoT data logging on blockchain network
5. NFT marketplace for digital asset trading
6. DeFi lending and borrowing platform
7. Decentralized identity management system
8. Healthcare data sharing using blockchain
9. Cross-border payments with stablecoins
10. Blockchain-based crowdfunding platform
11. Peer-to-peer energy trading using blockchain



Tutorials

Course Outcomes (COs)

CO1: Apply the core principles of blockchain, distributed ledgers, and cryptographic foundations to design basic blockchain-based solutions.

CO2: Analyze consensus mechanisms and blockchain security models used in public and permissioned networks.

CO3: Demonstrate the use of major blockchain platforms like Bitcoin, Ethereum, and Hyperledger Fabric.

CO4: Develop and deploy smart contracts and decentralized applications using industry-relevant tools.

CO5: Evaluate real-world blockchain applications and assess integration challenges with IoT, AI, and existing systems.

CO6: Assess scalability solutions, interoperability approaches, and emerging trends in blockchain ecosystems.

CO-PO Mapping

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	1	-	2	-	-	1	-	-	-	1	3	2	1	-
2	3	3	1	2	2	1	-	2	-	-	-	1	2	3	1	1
3	3	2	2	1	3	-	-	1	-	1	-	1	3	3	2	1
4	3	2	3	2	3	-	-	2	1	1	1	1	3	2	3	2
5	2	3	2	2	2	2	1	2	-	1	-	2	2	3	2	3
6	2	3	2	2	2	2	1	2	-	1	-	3	2	3	2	3
Avg	2.67	2.50	1.83	1.50	2.33	0.83	0.33	1.67	0.17	0.67	0.17	1.50	2.50	2.67	1.83	1.67

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L4, CO3 – L4, CO4 – L5, CO5 – L3, CO6 – L5

Future Courses Mapping: Advanced Blockchain , Web3 Technologies ,Cyber Security

Job Mapping: Blockchain Developer ,Web3 Developer ,Smart Contract Developer ,Hyperledger Developer .Blockchain Security Engineer

Textbook References

1. A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder; *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*; 1st Edition; Princeton University Press; 2016.
2. A. M. Antonopoulos; *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*; 2nd Edition; O'Reilly Media; 2017.
3. A. M. Antonopoulos, G. Wood; *Mastering Ethereum: Building Smart Contracts and DApps*; 1st Edition; O'Reilly Media; 2018.
4. D. Drescher; *Blockchain Basics: A Non-Technical Introduction in 25 Steps*; 1st Edition; Apress; 2017.
5. N. Gaur, L. Desrosiers, V. Ramakrishna, P. Novotny, A. Singh, M. Bocarsly; *Hyperledger Fabric In Action*; 1st Edition; Manning Publications; 2020.

MOOC References

1. A. Narayanan, J. Bonneau; *Bitcoin and Cryptocurrency Technologies*; Coursera; https://www.coursera.org/learn/cryptocurrency
2. University at Buffalo; *Blockchain Basics*; Coursera; https://www.coursera.org/learn/blockchain-basics
3. Prof. Sandip Chakrborty and Prof. Shamik Sural ; *Blockchain and its Applications*; NPTEL; https://nptel.ac.in/courses/106105235
4. Stipen Grider; *Ethereum and Solidity: The Complete Developer's Guide*; Udemy; https://www.udemy.com/course/ethereum-and-solidity-the-complete-developers-guide/
5. Linux Foundation; *Hyperledger Fabric Fundamentals*; edX; https://www.edx.org/learn/hyperledger

Syllabus Template **FF No. : 654**

MD***:: Ethical Hacking

Credits: 3 **Teaching Scheme: Theory: 2 Hours / Week**
Tutorial: 1 Hours / Week

Prerequisites:

Programming Fundamentals, Data Structures, Operating Systems, Computer Networks, Basic Cryptography

Course Objectives:

1. Understand ethical hacking methodologies and cybersecurity principles.
2. Learn reconnaissance, scanning, enumeration, and vulnerability assessment techniques.
3. Perform penetration testing on systems and networks in a controlled environment.
4. Identify web application and wireless security vulnerabilities.
5. Apply defensive techniques and security best practices.
6. Understand legal, ethical, and professional responsibilities in cybersecurity.

Course Outcomes:

1. Explain the fundamentals of ethical hacking, cybersecurity concepts, cyber threats, attack vectors, hacking methodologies, and legal and ethical issues (L2 – Understand)
2. Apply reconnaissance, footprinting, scanning, enumeration, and vulnerability assessment techniques using industry-standard security tools. (L3 – Apply)
3. Analyze system, network, web application, and wireless security vulnerabilities to identify potential threats and security weaknesses. (L4 – Analyze)
4. Demonstrate the use of penetration testing tools and techniques in a controlled laboratory environment to assess the security of computer systems and networks. (L3 – Apply)
5. Evaluate security incidents, analyze penetration testing results, and recommend appropriate mitigation and security hardening measures. (L5 – Evaluate)
6. Prepare professional penetration testing reports and communicate security findings while adhering to ethical standards, legal requirements, and responsible disclosure practices. (L5 – Evaluate)

Section 1: Topics/Contents

Unit-I Introduction to Ethical Hacking (6 Hours) **(6 Hours)** Information Security Fundamentals, Cyber Threat Landscape, Types of Hackers, Ethical Hacking Methodology, Penetration Testing Lifecycle, Cyber Kill Chain, CIA Triad,

Security Policies, Vulnerability, Threat, Risk, Legal and Ethical Issues, Cyber Laws, Responsible Disclosure

Unit II: Footprinting, Reconnaissance and Scanning (5 Hours) Passive Reconnaissance, Active Reconnaissance, Open Source Intelligence (OSINT), WHOIS, DNS Enumeration, Email Enumeration, Google Dorking, Network Scanning, Port Scanning, Banner Grabbing, Service Enumeration, Vulnerability Scanning

Unit III: System Hacking and Network Security (5 Hours) Password Attacks, Password Cracking Techniques, Authentication Attacks, Privilege Escalation Concepts, Malware Overview, Sniffing, ARP Spoofing, Man-in-the-Middle Attack, Session Hijacking, Network Security Controls, Firewalls, IDS/IPS Basics

Unit IV: Web Application and Wireless Security (6 Hours) Web Security: HTTP/HTTPS, OWASP Top 10, SQL Injection, Cross-Site Scripting , (XSS), Cross-Site Request Forgery (CSRF), Authentication Vulnerabilities, File Inclusion, Directory Traversal, Security Headers

Wireless Security: Wi-Fi Architecture, WPA/WPA2/WPA3, Wireless Threats, Rogue Access Points, Wireless Security Best Practices

Unit V: Penetration Testing, Reporting and Defense (6 Hours) Penetration Testing Framework, Exploitation Concepts, Post Exploitation Overview, Maintaining Access (Concept Only), Log Analysis, Incident Response Basics, Security Hardening, Patch Management, Security Auditing, Penetration Testing Report Writing, Executive Summary, Risk Rating, Remediation

Software Required: Kali Linux, VirtualBox/VMware, Wireshark, Nmap, Burp Suite Community, Edition, OWASP Juice Shop, DVWA, Metasploit Framework, OpenVAS or Nessus Essentials, John the Ripper, Hashcat, Aircrack-ng

List of Tutorials:

1. Study the cybersecurity landscape, ethical hacking methodology, CIA Triad, Cyber Kill Chain, and recent cyber attack case studies.
2. Perform Open Source Intelligence (OSINT) and reconnaissance using WHOIS, DNS lookup, Google Dorking, and publicly available information.

3. Demonstrate network discovery and port scanning using Nmap to identify hosts, open ports, services, and potential security risks.
4. Perform vulnerability assessment using OpenVAS or Nessus Essentials and interpret the results to recommend appropriate mitigation measures.
5. Demonstrate password auditing concepts using John the Ripper and Hashcat to evaluate password strength and password-cracking feasibility.
6. Capture and analyze network traffic using Wireshark to identify network protocols, suspicious packets, and potential security threats.
7. Analyze common web application vulnerabilities using OWASP Juice Shop or DVWA and discuss preventive security measures.
8. Study wireless network security concepts, Wi-Fi security protocols, attack scenarios, and recommended defense mechanisms using Aircrack-ng.
9. Explore the penetration testing lifecycle using Metasploit Framework in a controlled environment and study exploitation, post-exploitation, and system hardening techniques.
10. Prepare a professional penetration testing report including vulnerability analysis, CVSS-based risk ratings, remediation recommendations, and responsible disclosure practices.

List of Course Projects

1. Enterprise Network Vulnerability Assessment and Security Audit.
2. Web Application Penetration Testing using OWASP Juice Shop or DVWA.
3. Secure E-Commerce Website Security Assessment.
4. Wireless Network Security Analysis and Hardening.
5. Phishing Detection and Email Security Assessment.
6. Security Assessment of an IoT-Based Smart Home Network.
7. Linux Server Security Hardening and Vulnerability Analysis.
8. Cybersecurity Risk Assessment and Compliance Audit for a Small Organization.
9. Active Directory Security Assessment and Hardening
10. Secure Network Infrastructure Assessment for an Educational Institution
11. Security Assessment of a Cloud-Hosted Web Application
12. Incident Response and Digital Forensics for a Simulated Cyber Attack

List of Course Seminar Topics:

1. Ethical Hacking Methodology and Penetration Testing Lifecycle.
2. Cyber Kill Chain and MITRE ATT&CK Framework.
3. Open Source Intelligence (OSINT) Techniques.
4. Password Attacks and Authentication Security.
5. Vulnerability Assessment vs Penetration Testing.
6. OWASP Top 10 Web Application Vulnerabilities.
7. Wireless Security and Wi-Fi Attacks.
8. Network Sniffing, ARP Spoofing, and Man-in-the-Middle Attacks.
9. Security Hardening and Incident Response.
10. Artificial Intelligence in Cybersecurity.
11. Zero Trust Security Architecture.
12. Cloud Security Challenges and Best Practices.
13. Ransomware Trends and Defense Strategies.
14. Responsible Disclosure and Bug Bounty Programs.

List of Design-Based Home Assignments:

1. Design a Cyber Kill Chain for a real-world cyber attack and identify possible defense mechanisms at each stage.
2. Design a secure password policy and multi-factor authentication framework for an organization.
3. Design a secure enterprise network architecture incorporating firewalls, IDS/IPS, DMZ, and VPN.
4. Design a web application security checklist based on the OWASP Top 10.
5. Design a vulnerability assessment and penetration testing (VAPT) methodology for a small organization.
6. Design a wireless network security plan for a university campus.
7. Design an incident response workflow for handling ransomware attacks.

List of Case Study-Based Home Assignments:

1. WannaCry Ransomware Attack.
2. Equifax Data Breach.
3. SolarWinds Supply Chain Attack.
4. Colonial Pipeline Ransomware Attack.
5. MOVEit Transfer Vulnerability (CVE-2023-34362).
6. Log4Shell (Log4j) Vulnerability.
7. Uber Data Breach and Social Engineering Attack.

List of Blog-Based Home Assignments:

1. Ethical Hacking Methodology.
2. Open Source Intelligence (OSINT).
3. Password Cracking Techniques and Password Security.
4. Network Scanning and Enumeration.
5. OWASP Top 10 Web Application Vulnerabilities.
6. Wireless Network Security.
7. Penetration Testing Lifecycle.
8. Responsible Disclosure and Bug Bounty Programs.
9. Emerging Trends in AI and Cybersecurity.

List of Survey-Based Home Assignments:

1. Survey of Vulnerability Assessment Tools (OpenVAS, Nessus, Qualys).
2. Survey of Network Scanning Tools (Nmap, Masscan, Angry IP Scanner).
3. Survey of Password Auditing Tools (John the Ripper, Hashcat, Hydra).
4. Survey of Web Application Security Testing Tools (Burp Suite, OWASP ZAP, Nikto).
5. Survey of Wireless Security Assessment Tools (Aircrack-ng, Kismet, Wireshark).
6. Survey of SIEM and Log Analysis Tools.
7. Survey of Cybersecurity Frameworks (NIST CSF, ISO 27001, CIS Controls).

CO-PO Mapping:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	1	-	2	-	-	2	-	-	-	2	-	2	-	-
2	3	3	2	2	3	-	-	1	-	-	-	2	-	2	-	-
3	3	3	3	3	3	-	-	2	2	-	-	2	-	2	-	-
4	3	3	2	3	3	-	-	2	-	-	-	2	-	2	-	-
5	3	3	3	2	3	2	-	3	2	-	-	2	-	2	-	-
6	2	1	-	-	-	2	-	3	2	2	-	2	-	2	-	-
Avg	2.83	2.5	2.2	2.5	2.8	2		2.16		2		2		2		

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Text Books:

1. Rafay Baloch, **Ethical Hacking and Penetration Testing Guide**, CRC Press, 2017.
2. Matt Walker, **CEH Certified Ethical Hacker All-in-One Exam Guide**, McGraw-Hill Education, 5th Edition, 2024.
3. Jon Erickson, **Hacking: The Art of Exploitation**, 2nd Edition, No Starch Press, 2008.
4. William Stallings and Lawrie Brown, **Computer Security: Principles and Practice**, 5th Edition, Pearson, 2023.

Reference Books

1. **Dafydd Stuttard and Marcus Pinto**, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, 2nd Edition, Wiley, 2011.
2. **Georgia Weidman**, *Penetration Testing: A Hands-On Introduction to Hacking*, No Starch Press, 2014.
3. **David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni**, *Metasploit: The Penetration Tester's Guide*, No Starch Press, 2011.
4. **William Stallings**, *Network Security Essentials: Applications and Standards*, 7th Edition, Pearson, 2021.

For MOOCs and other learning Resources

1. https://onlinecourses.nptel.ac.in/
2. https://swayam.gov.in/
3. https://portswigger.net/web-security
4. https://owasp.org/www-project-juice-shop/

Syllabus Template **FF No. : 654**

MDM***: Augmented and Virtual Reality

Credits: 3 **Teaching Scheme: Theory: 2 Hours / Week**
Tutorial: 1 Hours / Week

Course Prerequisites: Basic Computer Fundamentals, Basic Programming Skills (C/C++), Mathematics (Linear Algebra, Geometry, Trigonometry), Fundamentals of Computer Graphics.

Course Objectives:

1. To introduce fundamental concepts of Virtual Reality, Augmented Reality, and Mixed Reality.
2. To understand hardware components, displays, sensors, and tracking mechanisms in AR/VR systems.
3. To apply basic geometric transformations and rendering concepts in virtual environments.
4. To explore fundamental computer vision and tracking techniques used in AR.
5. To design and develop simple interactive AR/VR applications using modern tools.
6. To understand applications and emerging trends in AR/VR/MR technologies.

Course Outcomes:

The student will be able to –

1. Explain fundamental concepts and applications of AR, VR, and MR systems.
2. Describe hardware components, displays, sensors, and tracking techniques in AR/VR.
3. Apply basic transformations and understand rendering concepts in virtual environments.
4. Explain and implement basic AR tracking techniques such as marker-based and markerless tracking.
5. Design and develop simple interactive AR/VR applications using development tools.
6. Identify applications and recent trends in AR/VR/MR technologies.

Course Relevance:

This course equips students with essential skills in AR/VR technologies that are widely used in industries like gaming, healthcare, education, and retail. It aligns with emerging trends such as immersive computing and the metaverse, while providing practical knowledge in rendering, computer vision, and interaction design. The course also opens opportunities for careers, research, and innovation in rapidly growing AR/VR domains.

Syllabus
Theory

Section 1: Topics/Contents

Unit 1: Introduction to Virtual Reality and Augmented Reality (4 Hours)

Definition of VR, AR and MR, differences between AR, VR, and MR, evolution and trends, components of AR, VR systems, input-output and processing units, applications in gaming healthcare education retail

Unit 2: Hardware Components of AR and VR Systems (5 Hours)

Displays in AR and VR: Audio; Haptic; Visual, and Other Sensory Displays.
Visual Perception: Requirements and Characteristics Spatial Display Models.
Tracking and Sensors: Calibration; Registration; Stationary and Mobile Tracking; Sensor Fusion.

Unit 3: Representing and Rendering the Virtual & Augmented Worlds (5 Hours)

Geometric Models and Transformations in VR: Position; Orientation; Axis-Angle Rotation; Viewing Transformations, Human Vision and Eye Movements: Implications for VR and AR.
Visual Perception: Depth; Motion; Color; Combining Sources, Visual Rendering Techniques

Section 2: Topics/Contents

Unit 4: Computer Vision, Tracking and AR Techniques (5 Hours)

Computer Vision for AR: Marker Tracking, Natural Feature Tracking, Outdoor Tracking, Marker-Based Tracking, Marker less Tracking
SLAM; AR Software Components and Content Creation Tools.

Unit 5: Motion, Interaction & Devices in AR/VR (5 Hours)

Motion in Real and Virtual Worlds.
Interaction: Motor Programs; Locomotion; Manipulation; Social Interaction.
Audio: Physics of Sound; Human Hearing; Auditory Perception and Rendering; AR/VR Devices.

Unit 6: Applications and Trends in AR/VR/MR (4 Hours)

Introduction to Mixed Reality; Applications of Mixed Reality; Input and Output in Mixed reality; Computer Vision and Mixed Reality.

Laboratory

List of Experiments

1) Introduction to Unity environment including scene creation, GameObjects, and components
2) Implementation of transformations such as position, rotation, and scaling with basic interaction
3) Development of a simple VR environment with camera movement and navigation
4) Implementation of collision detection and scoring system in a VR scene
5) Development of a marker-based AR application using Unity and Vuforia
6) Implementation of interaction in AR such as object scaling, rotation, and placement
7) Implementation of UI elements such as score display, buttons, and text in AR/VR
8) Development of a simple VR mini-game (ball game or shooting game)
9) Creation of animation and event handling in Unity for interactive scenes
10) Development of a basic mixed reality scenario using available toolkits
11) Implementation of simple audio integration in VR environment

Course Project

List of Course Projects

1) AR-based educational application using marker-based tracking
2) VR-based simulation such as virtual tour or training environment
3) AR product visualization application
4) Simple VR game with scoring and interaction
5) AR navigation prototype using image targets
6) VR-based storytelling or interactive scene
7) VR training simulation for basic tasks such as equipment handling or safety training
8) AR application for object visualization using multiple markers
9) VR game with animation, audio, and user interaction
10) AR-based maintenance or instruction guide for simple equipment
11) Mixed Reality demo combining real-world view with virtual objects

CO-PO Mapping

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	-	1	-	-	-	-	-	-	1	2	1	-	-
CO2	3	2	1	-	3	-	-	-	-	-	-	1	2	2	-	-
CO3	3	3	2	1	3	-	-	-	-	-	-	1	3	2	-	-
CO4	3	3	2	2	3	-	-	-	-	-	-	2	3	3	-	1
CO5	2	3	3	2	3	-	-	-	2	2	-	2	3	3	2	3
CO6	2	3	2	3	3	-	-	-	-	-	1	3	3	3	2	3
Average	2.67	2.67	1.67	1.33	2.67	0	0	0	0.33	0.33	0.17	1.67	2.67	2.33	0.67	1.17

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L2, CO3 – L3, CO4 – L3, CO5 – L4, CO6 – L2

Future Courses Mapping:

Advanced Computer Graphics, Computer Vision, Machine Learning, Deep Learning, , Game Development, Robotics, Spatial Computing, 3D Reconstruction, Metaverse Technologies, Extended Reality (XR) and similar courses

Job Mapping:

This course prepares students for diverse roles such as AR/VR Developer, Unity or Unreal Developer, 3D Graphics Programmer, Computer Vision Engineer, and XR (Extended Reality) Developer. It also supports career paths in game development, simulation engineering, and immersive UI/UX design. With the growing demand for metaverse technologies, students can explore opportunities as Metaverse Developers or Research Engineers in AR/VR/MR domains across various industries.

Text Books:

1) Gerard Jounghyun Kim, “Designing Virtual Systems: The Structured Approach”, 2005.
2) Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, “3D User Interfaces, Theory and Practice”, Addison Wesley, USA, 2005.
3) Designing for Mixed Reality, Kharis O'Connell Published by O'Reilly Media, Inc., 2016, ISBN: 9781491962381

Reference Books:

1) Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016
2) Understanding Virtual Reality: Interface, Application and Design, William R Sherman and Alan B Craig, (The Morgan Kaufmann Series in Computer Graphics)”. Morgan Kaufmann Publishers, San Francisco, CA, 2002
3) Developing Virtual Reality Applications: Foundations of Effective Design, Alan B Craig, William R Sherman and Jeffrey D Will, Morgan Kaufmann, 2009.

For MOOCs and other learning Resources

1) https://nptel.ac.in/courses/106/106/106106138/
2) https://www.coursera.org/learn/introduction-virtual-reality
3) https://www.coursera.org/learn/ar
4) https://www.udemy.com/share/101XPi/

Syllabus Template **FF No. : 654**

MDM***: Advanced Computer Architecture

Credits: 3 **Teaching Scheme: Theory: 2 Hours / Week**
Tutorial: 1 Hours / Week

Course Prerequisites: Digital Logic Design,Computer Organization and Architecture,Data Structures,Operating Systems (Basics)

Course Objectives:

Students will be able to:

1. Understand advanced architectural techniques used in modern processors.
2. Analyze instruction-level, thread-level, and data-level parallelism
3. Evaluate memory hierarchy design and optimization techniques.
4. Study multicore, manycore, and heterogeneous computing architectures.
5. Examine performance enhancement mechanisms in contemporary processors.
6. Explore emerging trends in computer architecture for AI, cloud, and high-performance computing systems.

Course Outcomes:

The student will be able to –

1)Relate advanced processor architectures and performance enhancement techniques

2) Compare pipelined, superscalar, and out-of-order execution mechanisms.

3)Justify memory hierarchy designs and their impact on system performance.

4) Distinguish multicore, parallel, and heterogeneous computing architectures.

5)Recommend GPU-based and accelerator-based computing platforms for modern applications.

6) Evaluate emerging architectural paradigms for future computing systems.

Course Relevance:

Modern computing systems demand high performance, energy efficiency, and scalability. Advanced Computer Architecture provides the foundation for understanding processor design, memory organization, multicore systems, parallel architectures, and specialized accelerators used in AI, cloud computing, and high-performance computing. The course prepares students for careers in system design, embedded systems, processor development, HPC, and AI infrastructure.

Syllabus
Theory

Section 1: Topics/Contents

Unit 1:Processor Performance and Pipelined Architectures (5 Hours)

Review of Computer Architecture Fundamentals; Processor Performance Metrics; CPU Performance Equation; Instruction Set Architecture vs Microarchitecture; PipeliningFundamentals; Pipeline Hazards -Structural Hazards, Data Hazards, Control Hazards; Hazard Detection and Resolution; Pipeline Performance Analysis; Case Study of Modern Pipelined Processors

Unit 2:Instruction-Level Parallelism and Superscalar Architectures (5 Hours)

Concept of Instruction-Level Parallelism (ILP); Static and Dynamic Scheduling; Register Renaming; Speculative Execution; Branch Prediction Techniques; Superscalar Processors; Out-of-Order Execution; Performance Evaluation of ILP Architectures

Unit 3: Memory Hierarchy and Advanced Memory Systems (5 Hours)

Memory Hierarchy Design; Cache Memory Organization; Cache Mapping Techniques; Cache Replacement Policies; Write Policies; Multi-Level Cache Architectures; Cache Performance Metrics; Virtual Memory; Translation Lookaside Buffer (TLB); Memory Consistency Models; Emerging Memory Technologies

Section 2: Topics/Contents

Unit 4: Multicore and Parallel Architectures (5Hours)

Motivation for Parallel Computing; Multicore Architectures ; Thread-Level Parallelism ; Simultaneous Multithreading (SMT); Chip Multiprocessors (CMP); Shared Memory Architectures; Distributed Memory Architectures; Interconnection Networks; Performance Analysis of Multicore Systems; Introduction to Parallel Programming Models

Unit 5:GPU and Heterogeneous Computing Architectures (4 Hours)

Evolution of GPU Computing, GPU Architecture Fundamentals; SIMT Execution Model; GPU Memory Hierarchy; CPU-GPU Heterogeneous Systems; Accelerator-Based Computing; AI Accelerators and Tensor Processing Units; Performance Considerations in Heterogeneous Systems

Unit 6: Emerging Trends in Computer Architecture (4 Hours)

Energy-Efficient Computing; Green Computing Architectures; Cloud Data Center Architectures; Edge and Fog Computing Architectures; Neuromorphic Computing; Quantum Computing Fundamentals; Architectures for Artificial Intelligence; Future Directions in Computer Architecture

Syllabus
Laboratory

List of Tutorials

1) CPU Performance Evaluation and Benchmark Analysis
2) Pipeline Hazard Identification and Resolution
3)Pipeline Scheduling Exercises
4) Branch Prediction Analysis
5) Register Renaming Exercises
6) Cache Mapping Problems
7)Cache Performance Analysis
8) Virtual Memory and TLB Calculations
9) Multicore Performance Evaluation
10) Interconnection Network Analysis

Design-Based Assignments

1. Design a Five-Stage Pipelined Processor.
2. Design an Efficient Branch Prediction Unit.
3. Design a Multi-Level Cache Hierarchy for a Mobile Processor.
4. Design a Multicore Architecture for Real-Time Systems.
5. Design an AI Accelerator Architecture.

Case Study Assignments

1. ARM Cortex-A Series Architecture
2. Intel Core Processor Architecture
3. AMD Zen Architecture
4. NVIDIA Hopper Architecture
5. Apple Silicon Architecture

Survey-Based Assignments

1. AI Accelerators in Modern Computing
2. Energy-Efficient Processor Design
3. Neuromorphic Computing Systems
4. Cloud Data Center Architectures

5. Quantum Computing Architectures

Blog-Based Assignments

1. Evolution of Processor Architectures
2. Future of Multicore Systems
3. GPU Computing in AI
4. Challenges in Memory System Design
5. Sustainable Computing Architectures

Syllabus
Course Project

List of Course Projects

1)Pipeline Simulator Design

2)Cache Memory Simulator

3) Branch Predictor Implementation

4) Multicore Performance Evaluation Framework

5) GPU Architecture Modeling

6) Comparative Analysis of Modern Processor Architectures

CO-PO Mapping

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1			3													
CO2					3											
CO3							2									
CO4											1					
CO5												1				
CO6															3	
Average	0	0	3	0	3	0	2	0	0	0	1	1	0	0	3	0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

Future Courses Mapping:

High Performance Computing; Parallel Computing; Embedded Systems Design
GPU Computing; Artificial Intelligence Systems

Job Mapping:

Computer Architecture Engineer, Embedded Systems Engineer, Processor Design Engineer Performance Engineer. Systems Architect, Hardware Validation Engineer, AI Infrastructure Engineer, HPC Engineer

Text Books:

1. Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann
2. Computer Organization and Design, 5th Edition, Morgan Pearson Education, Second Edition.
3. Advanced Computer Architecture, McGraw-Hill.

Reference Books:

1.Modern Processor Design, McGraw-Hill.
2. Structured Computer Organization, Pearson
3.Computer Architecture and Parallel Processing, McGraw-Hill.
J

For MOOCs and other learning Resources

1. NPTEL Computer Architecture Courses
2. MIT Open Courseware – Computer Architecture

3.Computer Architecture Research Resources (HPCA/ISCA)

Syllabus Template **FF No. : 654**

MDM***: Data Science

Credits: 3 **Teaching Scheme: Theory: 2 Hours / Week**
Tutorial: 1 Hours / Week

Course Prerequisites:

1. Engineering Mathematics / Applied Mathematics
2. Fundamentals of Programming

Course Objectives:

1. Learn concepts, techniques and tools they need to deal with various facets of data science practice, including data collection and integration
2. Understand the basic types of data and basic statistics
3. Identify the importance of data reduction and data visualization techniques

Course Outcomes:

1. Explain the fundamental concepts of Data Science, statistical modeling, probability distributions, and data-driven decision-making.
2. Classify different types of data and apply appropriate statistical techniques to summarize, analyze, and interpret datasets.
3. Perform data preprocessing tasks including data cleaning, transformation, scaling, feature selection, and outlier detection to improve data quality.
4. Develop Python programs using lists, dictionaries, NumPy arrays, vectors, matrices, and Pandas DataFrames for data analysis and processing applications.
5. Apply the data pre-processing, design feature engineering pipelines, and evaluate machine learning models using Python to solve complex, data-driven engineering problems.
6. Visualize, Build and evaluate basic predictive models using regression techniques and interpret analytical results for decision-making.

Section 1: Topics/Contents

UNIT- I Introduction **(4 Hours)**

Definition of Data Science- Big Data and Data Science hype – and getting past the hype - Datafication- Current landscape of perspectives - Statistical Inference - Populations and samples – Statistical modelling, probability distributions, fitting a model – Over fitting.Python environment for data science.

UNIT- II Data Types & Statistical Description (5 Hours)

Types of Data: Attributes and Measurement, Attribute, The Type of an Attribute, The Different Types of Attributes, Describing Attributes by the Number of Values, Asymmetric Attributes, Binary Attribute, Nominal Attributes, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes.

Basic Statistical Descriptions of Data: Measuring the Central Tendency: Mean, Median, and Mode, Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range, Graphic Displays of Basic Statistical Descriptions of Data.

UNIT- III Data Cleaning and Preparation (4 Hours)

Types of Attributes Data Sources and Data Quality Data Cleaning Identifying Outliers Measures of Centre and Spread Data Exploration: Hypothesis Testing Vs. Exploratory Data Analysis Data Transformation, Structural Inspection, Data encoding.

UNIT- IV **Foundations of Data Science Programming with Python (4 Hours)**

Vectors: NumPy Arrays & Vectors Array Indexing and Slicing (Subsetting), Multi-dimensional Arrays ,Matrices (2D NumPy Arrays),PCA, Matrix Subsetting(array[row_slice, col_slice]) ,Categorical Data & Pandas DataFrames : (loc,iloc), Category Levels & Summaries, Extending DataFrames, Sorting. Python Lists & Dictionaries: Data Science Applications of Lists and Dictionaries, Analyze a CSV dataset using lists and dictionaries.Advanced Multidimensional Arrays & Vector Engineering (NumPy): Introduction to NumPy, Creating NumPy Arrays, Indexing, Slicing, and Reshaping, Vectorized Operations

UNIT- V Working with Data, Exploratory Analysis, and Feature Engineering (5 Hours)

Data Manipulation with Pandas: Introduction to Panda, Importing datasets, Data inspection(head(), tail(), info(), describe()),Data aggregation and grouping, Merging and concatenating DataFrames, Exploratory Data Analysis (EDA): Descriptive statistics, Distribution analysis, Correlation analysis, Detecting trends and patterns. Feature Engineering: Feature selection, Feature extraction, Feature scaling. Train-Test Split and Cross Validation: Training, validation, and testing datasets, Train-Test Split using Scikit-learn, Model Evaluation: Confusion Matrix, Accuracy, Precision, Recall.

UNIT- VI (5 Hours)

Data Visualization: 2D and 3D Visualization. Introduction, Chart: Chart Legend, Bar Chart, Box Plot, Histogram, Line Graph: Multiple Lines in Line Graph, Scatter, Core Visualizations (Matplotlib & Seaborn) Plot. Regression: Linear Regression Analysis, Multiple Linear regression.

Tutorial List:

1. Simulating Population vs. Sample Math and Data Type Identification
2. Calculate metrics of central tendency and dispersion manually, check data distribution, and understand using python.
3. Classify the attributes into appropriate categories
4. Data Cleaning and Pre-processing Techniques
5. Working with Vectors, Matrices, Arrays and Data Frames
6. Lists, Functions and Control Structures in pythons
7. Data Visualization using in Python
8. Linear Regression and Predictive Analytics using python

	Program Outcomes (PO)												**PSO**		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CS3215.1	3	2	1	2	2	-	-	1	-	1	-	2	3	2	2
CS3215.2	3	3	2	3	2	-	-	1	-	2	-	2	3	2	3
CS3215.3	2	3	3	3	3	-	-	1	1	2	-	2	3	3	3
CS3215.4	2	2	2	2	3	-	-	1	1	2	-	2	3	2	3
CS3215.5	2	3	3	3	3	1	-	1	2	2	1	2	3	3	3
CS3215.6	3	3	3	3	3	1	-	1	2	3	1	2	3	3	3
Average	**2.5**	**2.66**	**2.33**	**2.66**	**2.66**	**0.33**	**2**	**1**	**1.0**	**2.0**	**0333**	**2.0**	**3.0**	**2.5**	**2.83**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L1, CO2– L2, CO3 – L3, CO4 – L3, CO5 – L2 and CO6 – L4

Future Courses Mapping:

Machine Learning, AI Fundamentals, Data Analytics, Data Mining, Business Analytics

Job Mapping:

Data Analyst Trainee, Data Engineer Associate, Statistical Programmer

TEXTBOOKS:

1. Doing Data Science, Straight Talk from The Frontline. Cathy O'Neil and Rachel Schutt, O'Reilly,2014.
2. Data Analysis by Wes McKinney is O'Reilly Media

REFERENCE BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed. The Morgan Kaufmann Series in Data Management Systems.
2. Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education.
3. Python Data Science Handbook Jake VanderPlas
4. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, Aurélien Géron O'Reilly Media.

MOOCs and other learning Resources

1. www.nptelvideos.in,
2. https://developers.google.com/machine-learning/crash-course
3. https://classroom.volp.in/
4. https://www.coursera.org/professional-certificates/ibm-data-science

Syllabus Template **FF No. : 654**

CS3227: Design Thinking 5, 6

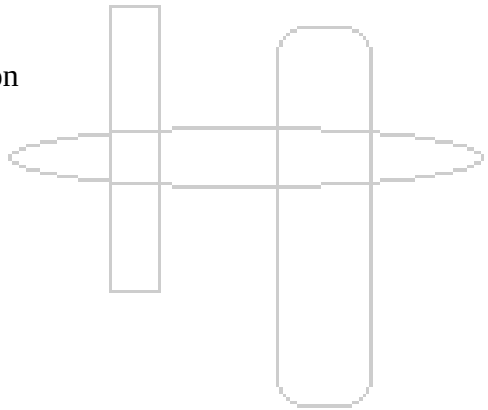
Credits: 01 **Teaching Scheme: Tutorial 01 Hr/week**

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:
To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship



Section 2: Topics/Contents

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

Course Outcome: **[**Publication of paper or patent]
The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	3
CO2	1	1	1	1	1	--	--	--	--	--	--	1	2	1	1	3
CO3	2	2	3	3	2	2	1	2	2	3	--	1	2	2	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	-	-	2	3
CO5	1	1	1	1	1	--	--	--	--	--	--	1	-	-	1	2
CO6	2	2	2	2	2	2	1	3	2	3	--	1	2	2	2	3
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	1
Average	**1.57**	**1.57**	**1.71**	**1.71**	**1.57**	**2.0**	**1.0**	**2.33**	**2.0**	**3.0**	**1.0**	**1.0**	1.66	1.66	1.71	2.5

Syllabus Template **FF No. : 654**

CS3225: Engineering Research And Innovation – I, II

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

Course Prerequisites: Problem Based Learning

Course Objectives:

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

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, 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 centric 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.

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, 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.

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 "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using 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).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.
- It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.

- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects can be taken up:

1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

...not limited to.....Faculty and students are free to include other areas which meet the society requirements at large.

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 Bloom's Taxonomy.

To focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

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 Robart Capraro, 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: www.nptelvideos.in

https://worldwide.espacenet.com/

Course Outcomes:

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

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

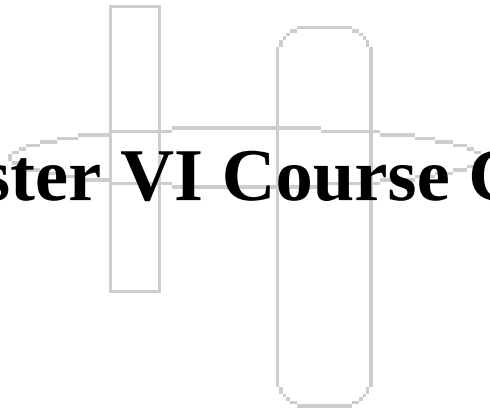
CO PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		3	3
CO6	2	2	3	3	2				3		3	2	3		3	3
Average	**2.0**	**2.0**	**2.83**	**2.83**	**2.6**	**2.5**	**2.0**	**2.0**	**3.0**	**1.0**	**2.16**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

T. Y. B. Tech. Computer Engineering
AY 2026-27



Semester VI Course Content

Syllabus Template **FF No. : 654**

CSXXXX:: Artificial Intelligence and Machine Learning

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**

Lab: 2 Hours/Week

Course Prerequisites: Basic programming Skills (C/C++/Java/Python), Data Structures, Design and Analysis of Algorithms, Mathematical Foundations of Computer Science.

Course Objectives:

1. To introduce intelligent agents, environment and problem formulation
2. To analyze and apply various searching strategies for solving AI problems
3. To develop predictive models using supervised machine learning techniques
4. To apply unsupervised algorithms
5. To understand and implement time series analysis
6. To understand and implement fundamentals of artificial neural networks

Course Outcomes:

The student will be able to –

1) Explain the concepts of intelligent agents, problem environments, knowledge representation, and problem formulation in Artificial Intelligence.

2) Apply uninformed search, informed search, game playing, and adversarial search algorithms to solve AI problems efficiently.

3) Develop predictive models using supervised machine learning techniques and evaluate their performance.

4) Apply unsupervised learning techniques for pattern discovery, clustering

5) Analyze temporal data using time series modeling and forecasting techniques

6) Design and implement Artificial Neural Network architectures

Course Relevance:

This is a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of computer engineering such as in Industries; research etc. as a basic prerequisite course. Data Structures are a crucial part of computer algorithms as they allow programmers to do data management efficiently. A wise selection of data structures can improve the performance of a computer program or algorithm in a more useful way.

Syllabus
Theory

Unit 1: Fundamentals of Artificial Intelligence (6 Hours)

Introduction: AI Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, Turing Test.

Environments: The nature of environments, PEAS.

Intelligent Agents: Types of agents- simple reflex, model based, goal based, utility agents.

Formulation of problems: E.g. Tic-Tac-Toe Game, Water-Jar problem, Missionaries and Cannibals' Problem, Vacuum World Problem, N-Queens Problem, Sudoku, 8-Puzzle Problem, Maze Problem, Route-Finding Problem, etc.

Unit 2: Problem Solving & Search Techniques (9 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.

Informed Search Methods: Hill Climbing, Best First Search, A* Algorithm, Game playing: Minimax Search, Alpha-Beta Cut offs.

Unit 3: Supervised Learning (10 Hours)

Basics of Machine Learning: supervised, unsupervised, reinforcement learning.

Supervised Learning Algorithms: Linear regression, logistic regression, k-nearest neighbors (k-NN), decision trees, support vector machines (conceptual), Ensembles: bagging, boosting.

Model evaluation: Train-test split, cross-validation; Performance Metrics: Accuracy, Precision, Recall, F1, ROC; overfitting, underfitting and basic techniques to eliminate those issues; bias-variance trade off.

Unit 4: Unsupervised Learning (5 Hours)

Clustering: K-means clustering, hierarchical clustering; **Dimensionality reduction:** PCA (intro); model evaluation.

Unit 5: Time Series Analysis and Forecasting (5 Hours)

Sequential vs time series analysis, Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA), Seasonal Auto Regressive Integrated Moving Average (SARIMA), model evaluation.

Unit 6: Introduction to Neural Networks (7 Hours)

Perceptron and basic neural networks, basics of hyperparameters, types of activation functions, types of optimizers, feedforward and backpropagation neural networks, model evaluation.

Syllabus
Laboratory

List of Experiments

1. Implement the uninformed search algorithms: Breadth First Search (BFS) and Depth First Search (DFS) for Water-Jar problem and Missionaries and Cannibals' Problem (C/C++/Java).
2. Implement the informed search algorithms: Best First Search and A* Search for 8-Puzzle problem and Route-Finding Problem (C/C++/Java).
3. Implement the supervised learning algorithm: k-NN (C/C++/Java. No use of readymade k-NN model)
4. Implement and evaluate Decision Tree model. (Empirical analysis of hyperparameter tuning using python)
5. Implement and compare Ensemble Learning using different classifiers or regressors (Empirical analysis of ensemble of minimum 5 models using python)
6. Implement the unsupervised learning algorithm: k-means clustering (C/C++/Java. No use of readymade k-means model)
7. Implement and evaluate time series models. (Empirical analysis of hyperparameter tuning using python)
8. Implement and evaluate neural network models. (Empirical analysis of hyperparameter tuning using python).

Syllabus
Course Project

List of Course Projects

Students must complete an end-to-end ML pipeline:

- Problem definition
- Data acquisition
- Preprocessing
- Model selection
- Empirical comparative performance analysis
- Interpretation

Suggested Projects (not limited to it):

1. Pattern Recognition
2. NLP
3. Edge AI

4. Recommendation Systems
5. AI-Based Gaming Applications
6. AIML in IoT, Robotics and Automation
7. AIML in Smart city analytics and management system
8. AIML in E-Commerce
9. AIML in Education
10. AIML in Disaster Management
11. Intelligent Logistics
12. Smart Agriculture
13. Smart Healthcare
14. Financial Forecasting
15. Weather Forecasting
16. Intelligent Energy Management System
17. Path finding using reinforcement learning
18. Sustainable AI
19. Explainable AI
20. Responsible and ethical AI

CO-PO Mapping

	Program Outcomes (PO)												PSO		
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2												2	
CO2	3	3		2	2							2	3	2	
CO3	2	3	2	2	3							2	3	2	2
CO4	2	3		3	3	2	2					2	3	2	2
CO5	2	3	2	2	3							2	3	2	2
CO6	2	3	3	2	3						2	2	3	2	2
Average	**2.33**	**2.83**	**2.33**	**2.2**	**2.8**	**2.0**	**2.0**				**2.0**	**2.0**	**3.0**	**2.0**	**2.0**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L2, CO2– L3, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L4

Future Courses Mapping:

Deep Learning, Natural Language Processing, Computer Vision, Generative Artificial Intelligence

Job Mapping:

The skills and knowledge acquired from this course will prepare students for following job roles: AI Engineer, Machine Learning Engineer, Data Scientist, Data Analyst, AI Research Associate, Predictive Analytics Engineer, NLP Engineer, Computer Vision Engineer, AI Consultant

Text Books:

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", Pearson Education, 4th Edition.
2. Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 3rd Edition, ISBN: 978-0-12-381479-1
3. Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, 4th Edition
4. Tom Mitchell, "Machine Learning", McGraw-Hill Science, ISBN: 0070428077

Reference Books:

For MOOCs and other learning Resources

1. www.nptelvideos.in,
2. www.geeksforgeeks.org

Syllabus Template **FF No. : 654**

CSXXXX:: Cryptography and Network Security

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

Lab: 2 Hours/Week

Course Prerequisites: Computer Networks

Course Objectives:

1. Apply cryptographic techniques and security protocols to secure systems and networks.
2. Identify, resolve, and mitigate programming bugs and cyber threats.
3. Design secure systems using blockchain technology and ensure application security.
4. Understand and apply cloud security and physical security principles.
5. Integrate AI in cyber security and develop business continuity and disaster recovery plans.
6. Implement ethical hacking practices and perform effective penetration testing.

Course Relevance:
Cyber Security teaches how to protect operating systems, networks, and data from cyber attacks, monitor systems, and mitigate threats, aiming to develop skills to prevent attacks and protect data privacy.

Section 1: Topics/Contents

Information security **(10 Hours)**

- **Key Security Properties:** Confidentiality, Integrity, Availability.
- **Risk Management:** Understanding governance policies, frameworks, laws, regulations, guidelines, and compliance.
- Mathematical background for cryptography: modulo arithmetic, GCD (Euclid's algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).
-
- **Symmetric Key Cryptography:** Role of random numbers and nonce in security, importance of prime numbers, GCD, Euclid's Algorithm, Extended Euclid's algorithm.
- **Data Encryption Standard (DES):** Block cipher, stream cipher (ex. RC4), Feistel structure, block cipher modes of operation, S-DES, attacks on DES, S-AES, AES.

- **Public Key Cryptography:** RSA algorithm, key generation, attacks on RSA.
- **Elliptic Curve Cryptography (ECC):** Elliptic curves over real numbers and Zp, elliptic curve arithmetic. Diffie Hellman Key exchange,

Network Security **(10 Hours)**

- **Certificates and Hashing:** Message authentication Code (MAC) and the digital signature. Key management, digital certificates and the Public Key Infrastructure (PKI), Properties of hash functions, HASH + SALT, hashing algorithms (SHA1, SHA512).
- **Authentication and Authorization:** Network access control (Kerberos, multifactor authentication).
- **Transport-Level Security:** Web security considerations, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS standard, Secure Shell (SSH) application, IPsec.
- **Application Security:** Security by design, writing secure code, static and dynamic application security testing (SAST and DAST), interactive application security testing (IAST), Integrated Security in DevOps, OWASP, Application Security Services,

Section 2: Topics/Contents

Cyber Attacks and Penetration Testing **(10 Hours)**

- **Cyber Ethics:** Threats, threat modeling, injections, sniffing, and types of attacks.
- **Security Vulnerabilities:** risk, attack types, countermeasures.
- **Protocol Vulnerabilities:** DoS and DDoS, session hijacking, ARP spoofing.
- **Software Vulnerabilities:** Phishing, buffer overflow, cross-site scripting attack, ransomware, SYN-flooding, SQL-injection, DNS poisoning.
- **Penetration Testing:** Difference from automated vulnerability scans, objectives and limitations of a pen test, scoping and planning pen tests, executing pen tests and managing findings. Introduction to SDL (Secure Development Lifecycle) – Merging Security into SDLC.

Network And Physical Security **(10 Hours)**

- **Physical Security:** Physical access types, crime prevention through environmental design (CPTED).
- **IoT Security:** Definitions of OT, IoT, IIoT, and ICS, most widely used protocols in IoT environments (MQTT and CoAP).
- **Business Continuity (BC):** RTP/RPO, RTO, MTPD, ISO 22301 standard for business continuity management, importance, differences between BCMS and DRMS, risk management, testing, maintenance., Operation Resilience,

Cloud Security : Principles / Key Concepts of Cloud Security: Overview of cloud security principles and key concepts.

Role of AI in Cyber Security: Examination of how AI is integrated into cyber security.

Challenges and Opportunities of AI in Cyber Security: Analysis of the challenges and opportunities presented by AI in the field of cyber security.

List of Practicals (Minimum Six)

1. Develop the program of Euclid's algorithm to find the GCD of two integers.
   Objective: Euclid's Algorithm is a classical method for finding the Greatest Common Divisor (GCD) or Greatest Common Factor (GCF) of two integers. The objective of Euclid's Algorithm is to provide an efficient and systematic way to compute the GCD, which is the largest positive integer that divides both numbers without leaving a remainder.
2. Develop the program to implement the DES algorithm for encryption and decryption. Assume a suitable key.
   Objective: The Data Encryption Standard (DES) is a symmetric-key block cipher designed to encrypt and decrypt data in a secure and efficient manner. Its primary objective is to provide a method of securing electronic communications and data storage.
3. Develop the program to implement the AES algorithm for encryption and decryption. Assume a suitable key.
   Objective: Apply symmetric-key cryptography concepts to ensure confidentiality of data.
4. Develop the program to implement the RSA algorithm for encryption and decryption. Assume suitable Private and Public Keys.
   Objective: RSA provides a way to encrypt sensitive data so that it can only be decrypted by the intended recipient. By using a pair of public and private keys, RSA ensures that even if the public key is exposed, the data remains confidential as only the corresponding private key can decrypt it.
5. Decoding Email Headers: A Practical Approach to Identifying and Mitigating Cyber Threats.
   Objective: To understand and analyse email headers to detect signs of phishing, spoofing, or other malicious activities. This assignment aims to equip students with practical skills for email forensics and cybersecurity analysis.
6. Unveiling the Power of Google Hacking: Techniques, Risks, and Ethical Implications
   Objective: To understand and utilize Google Hacking techniques for ethical purposes, such as vulnerability assessment, information gathering, and cybersecurity awareness. This assignment aims to teach students the responsible use of publicly accessible data and highlight the potential risks posed by improper use.
7. Exploring the Role and Implementation of SSL Certificates in Secure Communication.
   Objective: To understand the significance, creation, and practical usage of SSL certificates in ensuring secure web communications. This assignment will emphasize both theoretical knowledge and hands-on experience with SSL certificates.
8. Investigate Vulnerability Scanners:
   - Explore and analyze the following vulnerability scanners:
     - Nessus
     - OpenVAS (Greenbone Vulnerability Manager)
     - QualysGuard
   - Identify the types of vulnerabilities each scanner targets (e.g., network vulnerabilities, web application vulnerabilities, etc.).
   - Compare the features of each scanner, including detection accuracy, ease of use, and integration with other security tools.
   - Evaluate the strengths and limitations of each tool in different security scenarios.
9. Web Application Vulnerability Scanners:
   - Research the following web application scanners:

- Burp Suite
- OWASP ZAP (Zed Attack Proxy)
- Acunetix

- Analyse their capabilities in detecting vulnerabilities such as SQL injection, XSS (Cross-Site Scripting), and CSRF (Cross-Site Request Forgery).
  Discuss how these scanners fit into a penetration testing workflow and the benefits of automating web application vulnerability scans.

10. Network Vulnerability Scanners:
    - Examine Nmap (Network Mapper) and Nikto as network vulnerability scanners.
    - Investigate how Nmap is used for network discovery and vulnerability scanning.
    - Evaluate how Nikto is used for detecting web server vulnerabilities.

    Provide use cases where these scanners are most effective and discuss their limitations in identifying network vulnerabilities.
11. Log Manipulation Tools:
    - Investigate the role of log manipulation tools in cybersecurity investigations and system administration.
      - Metasploit Framework/Ollydbg.
      - BleachBit
      - ExifTool
    - Explore how these tools are used for log analysis, file manipulation, and removing metadata to conceal activities during penetration testing or forensic investigations.
      Compare the features and functionalities of each tool in terms of user-friendliness, log manipulation capabilities, and integration with other tools.
12. Anti-Forensic Tools:
    - Study the anti-forensic tools listed below and their role in covering up traces of digital footprints.
      - CCleaner
      - Chkrootkit
    - Explore the functionality of each tool in deleting files, cleaning system traces, and scanning for rootkits.

    Discuss ethical concerns related to the use of anti-forensic tools and their potential misuse in criminal activities or to circumvent digital investigations.
13. Implementation of SHA.

Text Books: ***(As per IEEE format)***

- *William Stallings, "Cryptography and Network Security-Principles and Practices" 6th Edition, Pearson Education, 2014, ISBN13:9780133354690.*
- *Bernard Menezes, "Network Security and Cryptography", 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1.*
- *Raef Meeuwisse, "Cybersecurity for Beginners", 2nd Edition, Cyber Simplicity, 2017, ISBN-9781911452157*

Reference Books: ***(As per IEEE format)***

- *M. Speciner, R. Perlman, C. Kaufman, "Network Security: Private Communications in a Public World", Prentice Hall, 2002*

- *Michael Gregg, "The Network Security Test Lab: A Step-By-Step Guide", Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148.*
- *Matt Bishop, "Computer Security: Art and Science", 1st Edition, Pearson Education, 2002, ISBN 0201440997.*
- *Charlie Kaufman, Radia Perlman and Mike Spencer, "Network security, private communication in a public world", 2nd Edition, Prentice Hall, 2002, ISBN 9780130460196.*
- *V.K. Pachghare, "Cryptography and Information Security", 2nd Edition, PHI, 2015, ISBN-978-81-203-5082-3.*

MOOCs Links and additional reading material:

- *Cryptography And Network SecurityBy Prof. Sourav Mukhopadhyay | IIT Kharagpur*Cryptography And Network Security - Course (nptel.ac.in)
- *Information Security and Cyber ForensicsBy Prof. Pratosh Bansal | Devi Ahilya Vishwavidyalaya, Indore*, Information Security and Cyber Forensics - Course (swayam2.ac.in)

LEARNING PLATFORMS

1) https://tryhackme.com/

TryHackMe is an online cybersecurity learning platform designed to help beginners and professionals develop practical security skills through hands-on labs and guided learning paths.

2) https://www.shodan.io/

Shodan is a specialized search engine that discovers and indexes internet-connected devices rather than traditional web pages. It helps cybersecurity professionals identify exposed systems, services, and potential security risks on the Internet.

Course Outcomes:

1. Explain the fundamental concepts of information security, security services, risk management, governance, compliance, and the mathematical foundations of cryptography used in cybersecurity (L2 – Understand)
2. Apply symmetric and asymmetric cryptographic techniques to ensure secure communication and data confidentiality (L3 – Apply)
3. Analyze authentication, authorization, hashing, digital signatures,application security protocols, and secure application development practices to protect networked systems.(L4 – Analyze)
4. Analyze cyber threats, network and software vulnerabilities, cyber-attacks, and penetration testing methodologies to identify security weaknesses and recommend appropriate countermeasures.(L4 – Analyze)
5. Evaluate security mechanisms for cloud computing, IoT, operational technology, physical security, business continuity, disaster recovery, and operational resilience to enhance organizational security.(L5 –Evaluate)

6. Evaluate the role of artificial intelligence in cybersecurity by analyzing its applications, opportunities, challenges, and ethical considerations for securing modern computing environments.(L5 – Evaluate)

CO-PO Mapping:

CO	Program Outcomes (PO)												PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	–	–	1	1	–	2	–	–	–	1	3		1	3
2	3	2	3	1	2	–	–	1	–	–	–	1				2
3	2	3	3	2	3	1	–	2	–	–	–	1			1	3
4	2	3	2	3	3	2	–	2	–	–	–	1		3	1	3
5	2	3	2	2	2	3	2	2	–	–	1	2				
6	2	3	2	2	3	2	1	3	–	1	–	3		3	3	3
Avg	2.67	2.4	2	2.33	1.8	1.5	2	0	1	1	1.5	2.67	2.4	2	2.33	1.8

Program Elective Courses (PEC1)

Course	Course Code
	Compiler Design
	UI/UX Design
	PCGPU
	Image Processing

Syllabus Template **FF No. : 654**

CS3053::Compiler Design

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**
Lab: 2 Hours/Week

Course Prerequisites: Operating System, Theory of Computer Science

Course Objectives:
1. Understand the process of program execution cycle.
2. Understand the translation process from High Level Languages to Machine Level Language.
3. Know the syntax and semantic analysis approaches for efficient code/program verification.
4. Learn the methods of code generation which helps for the optimization.
5. Learn code optimization and runtime code synthesis.
6. Know the process of compiler design for emerging programming languages.

Course Relevance:

All high level programming languages are easy for users to understand but not understandable to a computing machine. The computing machine knows only binary data. A translation is required to convert higher level language into machine level, so that the intended program can be executed. This translation is done by using a compiler. This course will give you detailed insights of how compilers function internally. This gives freedom to design your own programming language with its compiler. The key technology of Compiler Design principles is to provide an in-depth view of translation and optimization process. Compiler design covers basic translation mechanism and error detection and recovery.

Section 1: Topics and Contents

Unit-I: Introduction to Compilers, interpreters, Assembler, Linker and Loader [04 Hours]

Compilers: Introduction to compiler phases, features of machine-dependent and independent compilers, overview of types of compilers, introduction to cross compiler, Interpreters: compiler vs. interpreter, phases, and working, Preprocessor: header file and macro expansion, Assembler: Introduction to Assembler, overview of types of Assembler, Linker and Loader: Introduction to Linker and Loader, overview of types of Linker and Loader.

Unit-II: Lexical Analysis and Syntax Analysis: **[04 Hours]**

Introduction to Compiler, Phases and Passes, Bootstrapping, Role of a Lexical Analyzer, Specification and Recognition of Tokens, LEX/FLEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers, Implementing Scanners, Operator Precedence Parsers.

Unit-III: Syntax Analysis and Semantic Analysis: [05 Hours]

Bottom-Up Parsing, LR Parsers: Overview of types of LR Parsers, Constructing LALR parsing tables, Introduction to YACC/BISON, Type Checking, and Type Conversion, Symbol Table Structure.

Section 2: Topics and Contents

Unit-IV: Syntax-Directed Translation and Intermediate Code Generation: [05 Hours]

Syntax-DirectedDefinitions, Bottom-Up Evaluation, Intermediate Representations, and Intermediate Code Generation: various formats of intermediate codes, Error Detection & Recovery: Lexical Phase errors, syntactic phaseerrors, semantic errors. More about translation: Array references in arithmetic expressions.

Unit-V: Code Generation: [04 Hours]

Issues in Code Generation, Basic Blocks and Flow Graphs, Next-use information, Simple Code Generator, DAG representation of Basic Blocks, Peephole Optimization. Generating code from DAGs

Unit-VI: Code Optimization, Run-Time Environments and Data Flow Analysis: [06 Hours]

Code Optimization and Run-Time Environments: Principle Sources of Optimization, Optimization of basic blocks, Global Data Flow Analysis, RuntimeEnvironments, and Source Language issues. Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing, Machine Dependent Optimization, Data Flow Analysis**:** Constant propagation, live range analysis.

Case studies: LLVM compiler Infrastructure, compiling OOP features, Compiling in multicore environment, Deep learning compilation, Parallel Compilers, Web Compilers.

List of Tutorials (13):

1. Single and two pass Assembler
2. Two pass Macro processor
3. Types of Linkers
4. Types of Loaders
5. Examples on First and Follow
6. Examples on Lex/Flex regular expressions
7. Construction of LL(1) parser
8. Construction of SLR parsing table
9. Construction of Canonical LR parsing table
10. Examples on YACC/Bison grammar rules
11. Translation Scheme

12. Examples of Intermediate code generation by Quadruples
13. Examples of DAG representation
14. Examples of Basic Blocks and Next use information

List of Practical's (Minimum Six to be performed out of 10):

1. LEX/FLEX specification and programming regular expressions.
2. Implement LEX/FLEX code to count the number of characters, words and lines in an input file.
3. Implement LEX/FLEX code to select only lines that begin or end with the letter 'a' and delete everything else.
4. Convert all uppercase characters to lowercase except inside comments.
5. Change all numbers from decimal to hexadecimal notation, printing a summary statistic (number of replacements) to stderr.
6. Implement Lexical Analyzer for language C.
7. Implement LR/SLR/LALR Parser.
8. YAAC specifications and implement Parser for specified grammar.
9. Implement Parser for language C.
10. Implement Syntax directed Translator.
11. Implement an Intermediate code generator (three address code and Quadruples)
12. Implement a code optimizer for C/C++ subset.
13. Implement a code generator for C/C++ subset.

List of Course Project areas:

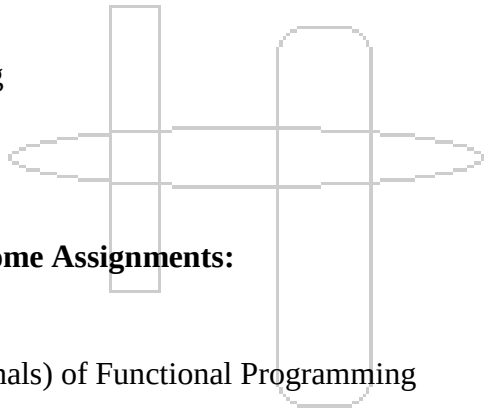
1. Compiler for subset of C using Lex and YAAC.
2. Compiler for Subset of Java programming Language.
3. Intermediate Code generator.
4. Code Optimizer.
5. Develop an Editor for Assembly programming. (Use available Assembler MASM/TASM to compile the code and execute in editor).
6. Design a system to check syntax and semantics of English Language.
7. Design a system to check syntax and semantics of a subset of Logical programming Language.
8. Design a System to check syntax and semantics of a subset of Python programming language.
9. Compiler for subset of C++ programming language.
10. Compiler for a subset of Algol programming language.

List of Seminar Topics:

1. Tools complementary to Lex
2. Tools complementary to YAAC
3. Semantic Analyser
4. Obsolete programming Language compiler advantage and issues
5. Android App program compiler
6. Approaches of Intermediate Code generation
7. Recent Trends in Compiler
8. Recent Trends in Interpreter
9. Decompilation
10. Compilation in multicore machines

List of Design based Home Assignments:

1. Recent methodologies in Intermediate Code Generator
2. Recent methodologies in Code Optimizer
3. Universal Compiler
4. Compiler for Deep learning
5. Recent trend in parsers



List of Case Study based Home Assignments:

1. Algol language Compiler
2. Compilation process (internals) of Functional Programming
3. Compilers for Mobile App development
4. LLVM compiler
5. Cross compiler
6. Types of open-source compilers

List of Blog based Home Assignments:

1. Decompilers: Ethical or Unethical?
2. Multi-paradigm programming compiler
3. State of the Art tools for rapid compiler development
4. Compiler for parallel machines
5. Compiler for distributed computing

List of Survey based Home Assignments:

1. Obsolete Programming Language Compilers
2. Obsolete Programming Language Interpreter
3. Compilers for various programming paradigms
4. Online compilers
5. Mobile app cross compiler

Text Books: ***(As per IEEE format)***

1. *Aho, A.V., Lam, M.S., Sethi, R., & Ullman, J.D. (2006). Compilers: Principles, Techniques,and Tools, Addison Wesley, ISBN 978-81317-2101-8 (2nd Edition).*

2. *Cooper, K., & Torczon, L. (2011). Engineering a compiler. Morgan Kaufmann, ISBN 155860-698-X.*

3. *Appel, A. W. (2004). Modern compiler implementation in C. Cambridge university press.*

4. *Appel, A. W., & Jens, P. (2002). Modern compiler implementation in Java. In ISBN 0–521–58388–8. Cambridge University Press.*

5. *Appel, A. W. (1998). Modern Compiler Implementation in ML, In ISBN 0-521-60764-.Cambridge University Press.*

6. Raghavan, V. (2010). Principles of Compiler Design. Tata McGraw-Hill Education

Reference Books: ***(As per IEEE format)***

1. Muchnick, S. (1997). Advanced compiler design implementation. Morgan Kaufmann, ISBN8178672413.

2. Levine, J. R., Mason, J., Levine, J. R., Mason, T., Brown, D., Levine, J. R., & Levine, P.(1992). Lex *& yacc. "O'Reilly Media, Inc".*

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. https://swayam.gov.in/nd1_noc20_cs13/preview
3. https://www.udacity.com/course/compilers-theory-and-practice--ud168
4. https://online.stanford.edu/courses/soe-ycscs1-compilers

Course Outcomes:

The student will be able to –

1. Design basic components of a compiler including scanner, parser, and code generator.
2. Perform semantic analysis in a syntax-directed fashion using attributed definitions.

3. Apply local and global code optimization techniques.
4. Synthesize machine code for the runtime environment.
5. Develop software solutions for the problems related to compiler construction.
6. Adapt themselves to the emerging trends in language processing.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO2	3	2	1	0	1	0	0	0	0	0	0	0	3	3	0	0
CO3	2	3	3	2	0	2	0	0	0	0	0	0	0	3	0	2
CO4	3	3	3	2	2	0	0	0	0	0	0	0	3	0	0	0
CO5	3	3	1	2	3	2	3	2	1	0	0	3	3	3	3	2
CO6	2	3	0	1	2	0	3	0	0	0	0	1	0	3	3	0
Average	2.67	2.83	2	1.75	2	2	3	2	1	0	0	2	3	3	3	2

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65

L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Course Mapping: Parallel Compiler

Job Mapping: Compiler Engineer

Syllabus Template **FF No. : 654**

CS3053:: UI/UX Design

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**
Lab: 2 Hours/Week

Course Prerequisites: Programming Skills & Software Engineering

Course Objectives:

1. To differentiate among concepts - Usability, User Interface, User Experience and HCI,
2. To define problem space; identifying User Categories and User Requirements utilizing interaction concepts,
3. To understand usage of Sketching, Wireframing and Prototyping in UI/UX design process,
4. To design a user-friendly user interface based on UI/UX design principles,
5. To apply usability evaluation methods on UI/UX designs to identify the usability issues,
6. To conduct the High-Fidelity Prototyping, Handoff, & Final Review process with UI/UX designs.

Course Outcomes:

The student will be able to –

1. Understand the definition and practicality of User Interface (UI) & User Experience (UX).
2. Differentiate multiple UX research methods and implement them.
3. Analyze research data to guide design decisions in UI design.
4. Create interface layout and navigation structures for digital systems.
5. Build wireframes and fidelity prototypes UI design principles.
6. Apply usability evaluation methods to create aesthetically sound and useful user interfaces. .

Course Relevance:

UI/UX design is a multi-disciplinary technical course with a goal of designing user-friendly mobile & web applications and their user interfaces. This is required as a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of Computer Sc / Engg., industries as well as product design & research. It aims to provide a customer-centric user experience to support a more accessible, useful & maintainable user interaction.

Syllabus
Theory

Section 1: Interaction Design Thoery

Unit 1: Foundation of HCI & UI Design (7 Hours)

Definitions - Human-Computer Interaction (HCI), Usability, User Interface (UI) and User Experience (UX), Interdisciplinary Nature, Measurable Human factors, Distinction between UI, UX, and Usability; Good UI/UX design and its Business Value. Double Diamond Design Process, Mapping with Software Development Life Cycle (SDLC), Data Analysis / Synthesis for Identification of Patterns and Pain Points.

Unit 2: Problem Space, User Requirements & Interaction Concepts (7 Hours)

Problem Statements / Definition with its Space, Categorization of Users: Demographic & Modern Approach, User Persona, User Requirements. Miller's Principle, Golden Rules of User Interface Design, Task Analysis, Contextual Inquiry, Work Models, User Flow, Card Sorting, Journey Mapping.

Unit 3: Sketching, Wireframing & Prototyping (6 Hours)

Ideation Techniques & Sketching - Crazy Eights / Design Sprints, Wireframing & Related Tools, Industry-Standard Software (e.g., Figma, Sketch, Adobe etc,), Building Functional & Screen-by-Screen Layouts, Colors/Fonts/Logos/Branding), Prototyping, Types of Prototypes & Usability Testing Setup.

Section 2: Design of UI/UX Prototypes

Unit 4: Visual Design & User Interfaces (7 Hours)

Visual Design Principles - Gestalt Principles, Visual Hierarchy, Proximity, Alignment and Contrast, Accessible Color Palettes (WCAG Standards), Font/Color Pairings, Design Concept, Process of Design, Accessibility (A11y) Best Practices, Screen Reader Compatibility, Ethnographic Observations, Internationalization.

Unit 5: Usability Evaluation & Testing (7 Hours)

Expert-based Evaluations, User-based Evaluations, Formative Evaluations, Summative Evaluations, Heuristic Evaluations, Cognitive Walkthrough, Semiotic Analysis, User Surveys, Interviews, Usability Testing, Ethical Considerations, Test Script Creation, User Recruitment, Test Data Analysis, Affinity Maps.

Unit 6: High-Fidelity Prototyping, Handoff, & Final Review (6 Hours)

Creating High-Fidelity Designs, Pixel-Perfect Mockups, Animations and Micro-Interactions, Designer-Developer Handoff, Preparation of Design Files, Specs & Assets for Engineering Teams, Project Presentation, Reviews, High-Fidelity Prototype, Research Insights.

Syllabus
Laboratory

List of Experiments

1) To define a problem with related user types; along with user requirements.
2) To create an initial prototype UI with a sketch / wireframe.
3) To implement screen layout using a suitable UI design tool.
4) To refine the UI/UX design in relevance to interaction concepts..
5) To evaluate the UI design using a suitable usability evaluation method.
6) To present the designed UI prototype with relevant approach and justification.

Syllabus
Course Project

List of Sample Course Projects:

Design & evaluate -

1) Mobile app UI for expenses splitter
2) Mobile app for navigation support to visually impaired users
3) Web application for tracking agricultural / crop damages after the disasters
4) Mobile app to facilitate the reading habit among the school children
5) Mobile app to identification of fake currency
6) Web design to monitor share market trends & related investments

CO-PO Mapping

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	POII	PO12	PSO1	PSO2	PSO3	PSO4
PEC1.1	2	3										2				2
PEC1.2	2	3		2								2				2
PEC1.3	3	2	3		2	3	2	2	2			2			3	2
PEC1.4	2	3	3	2								2			2	2
PEC1.5	3									2	1				2	2
PEC1.6	3	2	3	2	2	3	2	2	2						3	2
Average	2.5	2.6	3.0	2.0	2.0	3.0	2.0	2.0	2.0	2.0	1.0	2.0			2.5	2.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble - 0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

Future Courses Mapping:

Human-Computer Interaction, Usable Security & Intelligent User Interfaces

Job Mapping:

UI Designer, Software Engineer & Mobile App Developer

Text Books:

1. Ben Shneiderman, "Designing the User Interface", Third Edition, Pearson Education, ISBN 81-7808-262-4.
2. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human-Computer Interaction", Third Edition, Pearson Education, ISBN 81- 297-0409-9.

Reference Books:

1. Donald Norman, "The Design of Everyday Things", 2002 Edition, Basic Books, ISBN 100-465- 06710-7.
2. Wilbert Galitz, "The Essential Guide to User Interface Design", Second Edition, Wiley-Dreamtech India (P) Ltd., ISBN 81-265-0280-0.

For MOOCs and other learning Resources

1. www.nptelvideos.in
2. www.coursera.com
3. www.udemy.com

Syllabus Template **FF No. : 654**

CS3053:: Parallel Computing using GPU

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**
Lab: 2 Hours/Week

Course Prerequisites: Computer Organization and Architecture, Operating Systems, Data Structures, Design and Analysis of Algorithms, Programming in C/C++

Course Objectives:

1. To understand the fundamentals of parallel computing and parallel architectures.
2. To analyze performance metrics and scalability of parallel systems.
3. To develop parallel applications using CUDA programming.
4. To apply memory optimization and performance tuning techniques on GPUs.
5. TO utilize modern GPU architectures and CUDA libraries for high-performance computing.
6. To design and implement GPU-accelerated solutions for scientific, engineering, and AI applications.

Course Outcomes:
The student will be able to –

1. Relate parallel computing architectures, programming models, and performance metrics.
2. Utilize GPU architectures and memory hierarchy for efficient parallel execution.
3. Develop CUDA-based parallel applications using appropriate GPU programming constructs.
4. Apply synchronization, optimization, and profiling techniques to improve GPU application performance.
5. Choose modern GPU architectures, CUDA libraries, and parallel computing frameworks.
6. Propose GPU-accelerated solutions for scientific, engineering, AI, and real-world applications.

Course Relevance:

Parallel computing has become a fundamental technology for high-performance computing, artificial intelligence, machine learning, scientific simulations, data analytics, and cloud computing. Modern GPUs provide massive parallelism that accelerates computationally intensive applications. This course equips students with the knowledge and skills required to design, develop, optimize, and deploy parallel applications on contemporary GPU platforms.

Syllabus

Theory

Section 1: Foundations of Parallel Computing and CUDA

Unit 1: Basics of Parallel Computing (8 Hours)

Motivation and Need for Parallel Computing ; Concurrency vs Parallelism ; Levels of Parallelism ; Flynn's Classification of Computer Architectures ; Amdahl's Law and Gustafson's Law ; Shared Memory and Distributed Memory Architectures ;Interconnection Networks ;Introduction to High Performance Computing (HPC) ;Performance Metrics: Speedup, Efficiency, Throughput and Scalability , Basic GPU Architecture (Fermi)

Unit 2: Parallel Programming Models and CUDA Fundamentals (8 Hours)

Parallel Programming Paradigms ;Data Parallelism and Task Parallelism ;Introduction to CUDA ;CUDA Programming Model ;Host and Device Architecture ;Grids, Blocks and Threads ;Warp Execution Model ;Thread Scheduling ;Kernel Functions and Kernel Launch ; Thread Index Generation (1D, 2D, 3D) ;Device Functions and Error Handling

Unit 3:GPU Architecture and Memory Hierarchy (7 Hours)

Evolution of GPU Computing; Graphics Processing Pipeline; Streaming Multiprocessors (SMs); CUDA Cores and Tensor Cores; GPU Memory Hierarchy: Registers, Shared Memory, Global Memory, Constant Memory, Texture Memory; Unified Memory; Memory Coalescing; Occupancy and Resource Utilization; Overview of Volta, Ampere, Hopper and Blackwell Architectures

Section 2: CUDA Programming, Optimization and Real-World Applications

Unit 4: Parallel Algorithms using CUDA Programming (7 Hours)

Vector and Matrix Operations on GPU; Parallel Reduction; Matrix Multiplication; Optimization; Parallel Sorting Algorithms; Synchronization Mechanisms; Data Dependencies and Race Conditions; Concurrent Kernel Execution; CUDA Streams and Asynchronous Execution

Unit 5: Performance Optimization and CUDA Libraries (7 Hours)

Memory Optimization Techniques; Shared Memory Optimization; Bank Conflicts and Avoidance; Warp Divergence and Control Optimization; Occupancy Analysis; Profiling and Benchmarking; NVIDIA Nsight Systems; NVIDIA Nsight Compute; CUDA Libraries: Thrust, cuBLAS, cuFFT, cuDNN, Performance Evaluation and Tuning

Unit 6: Applications of Parallel Computing (6 Hours)

Scientific Computing Applications; Image and Video Processing; Big Data Analytics; AI and Machine Learning Workloads; Deep Learning Acceleration using GPUs; Medical Imaging **Applications:** Autonomous Systems and Robotics; Cloud-Based GPU Computing;

Case Studies: NVIDIA DGX Systems; Large Language Model Training; Weather Forecasting; Smart Healthcare Analytics

Syllabus
Laboratory

List of Experiments

1) Do performance Analysis using Amdahl's and Gustafson's Laws
2) To do CUDA Installation and Device Query
3) To implement Vector Addition using CUDA
4) To implement Matrix Addition using CUDA
5) To implement Matrix Multiplication using Global Memory
6) To implement Matrix Multiplication using Shared Memory
7) To implement Parallel Reduction Algorithm
8) To implement Prefix Sum (Scan) Implementation
9) To implement Image Processing using CUDA
10) To implement CUDA Streams and Concurrent Kernel Execution
11) To implement Parallel Sorting using Thrust Library
12) To do Profiling and Optimization using NVIDIA Nsight Tools

Syllabus
Course Project

List of Course Projects

1. GPU-Accelerated Image Processing Framework
2. Parallel Graph Analytics
3. Deep Learning Acceleration using CUDA
4. Smart Traffic Analytics using GPUs
5. GPU-based Medical Image Analysis
6. Real-Time Video Processing
7. Scientific Simulation using CUDA
8. Multi-GPU Performance Analysis

CO-PO Mapping

	Program Outcomes (PO)											PSO		
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1			3											
CO2					3									
CO3							2							
CO4											1			
CO5														
CO6														3
Average	**0**	**0**	**3.0**	**0**	**3.0**	**0**	**2.0**	**0**	**0**	**0**	**1.0**	**0**	**0**	**3.0**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2– L4, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Courses Mapping:

High Performance Computing, Distributed Systems, GPU Computing, Machine Learning Systems, Cloud Computing, Advanced Parallel Algorithms

Job Mapping:

CUDA Developer, GPU Software Engineer, HPC Application Developer, AI Infrastructure Engineer, Parallel Computing Engineer, Scientific Computing Engineer, Performance Optimization Engineer, Cloud GPU Engineer, Research Engineer (HPC and AI)

Text Books:

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-meiHwuCUDA: Programming Massively Parallel Processors: A Hands-On Approach. © ELSEVIER Inc. 3Jason Sanders and Edward KandrotCUDA by Example: An Introduction to General-Purpose GPU Programming"

Reference Books:

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

For MOOCs and other learning Resources

www.developer.nvidia.com

Syllabus Template **FF No. : 654**

CS3053:: Image Processing

Credits: 4 **Teaching Scheme Theory: 3 Hours/Week**

Lab: 2 Hours/Week

Course Prerequisites: Digital logic Design, Microprocessor, Computer Organization.

Course Objectives:

1. To describe different color models and image processing techniques.
2. To analyze image condition and deduce enhancement algorithms.
3. To apply image segmentation to identify the region of interest
4. To develop an algorithm to recognize the specified objects in the given image.
5. To study different image morphological operation.
6. To learn different image compression techniques.

Course Outcomes:

The student will be able to –

1) Recognize different color models and image processing techniques.
2) Select image enhancement algorithm to improve the quality of image.
3) Build image segmentation techniques to identify region of interest.
4) Predict image morphological techniques to resize the image.
5) Construct an algorithm to recognize the specified objects in the given image.
6) Identify different image compression techniques to reduce the size of image.

Course Relevance:

Image processing, inspired by the dominance of human vision, is a fundamental technology in modern intelligent systems and Industry 4.0.It enables machines to acquire, analyze, and interpret visual data for tasks such as object detection, feature extraction, and decision-making. In industrial automation, it is widely used for inspection, defect detection, process monitoring, and quality assurance to improve efficiency and accuracy. In multimedia and entertainment, advanced techniques support high-resolution formats like 4K through efficient compression and high frame rate processing. In social media and digital platforms, it enables image enhancement, filtering, and content-aware editing for better user engagement. Thus, image processing is a critical interdisciplinary field driving innovations in automation, artificial intelligence, and digital communication.

Syllabus

Theory

Section 1: Image fundamentals and Enhancement Techniques

Unit 1: Introduction of Image Processing (8 Hours)

Elements of an Image Processing System, Image Formation, Scenes and Digital Images
Basics of Vector Algebra in Image Processing, Human Visual System (HVS) and Perception
Color Vision and Color Models: RGB, HSV, YUV, CMYK, YCbCr, Relationships between Pixels (Adjacency, Connectivity, Distance Measures), Linear and Non-linear Operations on Images, Image Sampling and Quantization

Unit 2: Image Enhancement (8 Hours)

Introduction to Image Enhancement Techniques, Memory-less (Point Processing) Operations,
Spatial Domain Techniques, Image Denoising Filters (Mean, Median, Gaussian Filters),
Smoothing Operations, Sharpening Techniques (Laplacian, High-pass Filters) , Contrast Stretching and Enhancement , Histogram Processing and Histogram Equalization

Unit 3: Image Segmentation (8 Hours)

Fundamentals and Classification of Segmentation Techniques ,Edge-based Segmentation Methods ,Region-based Segmentation Techniques (Region Growing, Splitting & Merging) , Binarization Techniques: Global Thresholding, Adaptive Thresholding ,
Edge Detection Techniques: Derivative Filters, Sobel Operator, Canny Edge Detector,
Edge Linking and Boundary Detection,

Section 2: Morphology, Feature Analysis and Compression

Unit 4: Morphological Operations (8 Hours)

Introduction to Mathematical Morphology, Binary Morphology Concepts, Structuring Elements, Basic Operations: Erosion and Dilation, Compound Operations: Opening and Closing, Applications of Morphological Operations

Unit 5: Feature Extraction and Object Recognition (8 Hours)

Introduction to Feature Extraction, Feature Points and Detection (Line, Circle, Corner)
Line Detection: RANSAC, Hough Transform, Corner Detection: Harris Corner Detector
Feature Descriptors and Descriptor Matching, SIFT (Scale-Invariant Feature Transform)
Boundary Representation (Chain Code), Boundary Detection Techniques, Basics of Object Recognition

Unit 6: Image Compression (8 Hours)

Introduction and Need for Image Compression,
Image Redundancy Concepts Classification: Lossless and Lossy Compression, Run Length Encoding (RLE) , Huffman Coding , Shannon-Fano Coding ,JPEG Compression Technique ,
Performance Metrics (Compression Ratio, PSNR)

Syllabus
Laboratory

List of Experiments

Category 1: Basic Operations & Matrix Handling

- **Assignment 1: Image I/O and Matrix Exploration**
  Develop a program to load an image, display it, and export it in a different file format (e.g., .jpg to .png). Analyze and print the image dimensions, data type, and the raw matrix values of a specific 10×10 pixel window.
- **Assignment 2: Pixel-Level Arithmetic and Logic**
  Implement pixel-wise addition and subtraction between two images. Observe the effects of "clipping" versus "wrapping" and use image subtraction to detect the differences between two nearly identical frames.
- **Assignment 3: Image Sampling and Interpolation**
  Downsample an image by factors of 2, 4, and 8. Then, reconstruct the original size using **Nearest Neighbor** and **Bilinear Interpolation** to evaluate the loss of spatial resolution.
- **Assignment 4: Gray-Level Quantization**
  Reduce the number of intensity levels in a grayscale image (from 8-bit down to 1-bit/binary). Analyze the appearance of "false contouring" as the quantization levels decrease.

Category 2: Color & Intensity Transformations

- **Assignment 5: Color Space Conversion**
  Write a script to convert an RGB image into Grayscale, HSV, and YCbCr color spaces. Extract and display individual channels (e.g., the 'Hue' channel or 'Chrominance' channels) to understand their specific data representation.
- **Assignment 6: Point Processing and Negatives**
  Apply linear and non-linear intensity transformations, including brightness adjustment, contrast stretching, and the digital negative transformation ($s = L - 1 - r$).
- **Assignment 7: Histogram Generation and Analysis**
  Compute and plot the intensity histogram of various images (low-contrast, high-contrast, dark, and bright). Analyze how the distribution of pixels relates to visual quality.
- **Assignment 8: Histogram Equalization (HE)**
  Implement the Histogram Equalization algorithm to automatically enhance image contrast. Compare the original and processed histograms to verify a more uniform distribution.

Category 3: Filtering & Enhancement

- **Assignment 9: Image Smoothing and Noise Reduction**
  Apply **Mean**, **Gaussian**, and **Median** filters to an image corrupted with Salt-and-Pepper noise. Determine which filter best preserves edges while removing outliers.
- **Assignment 10: Image Sharpening in the Spatial Domain**
  Enhance the fine details of an image using the **Laplacian operator** and **High-boost filtering**. Compare the sharpened result against the original to identify artifacts or noise amplification.

Category 4: Segmentation & Edge Detection

- **Assignment 11: Global and Adaptive Thresholding**
  Segment an image into foreground and background using a fixed global threshold. Compare this against **Otsu's method** or adaptive thresholding for images with uneven lighting.
- **Assignment 12: Gradient-Based Edge Detection**
  Implement the **Sobel** and **Prewitt** operators to calculate gradient magnitude and direction. Generate an edge map and discuss the sensitivity of these operators to noise.
- **Assignment 13: The Canny Edge Detector**

Apply the multi-stage Canny algorithm, including Gaussian blurring, non-maximum suppression, and hysteresis thresholding, to produce thin, continuous edge lines.

- **Assignment 14: Region-Based Segmentation**
  Implement a **Region Growing** algorithm starting from a user-defined seed point. Evaluate how the similarity threshold affects the final segmented region.

Category 5: Morphological Operations

- **Assignment 15: Fundamental Erosion and Dilation**
  Perform erosion and dilation using various structuring elements (square, cross, disk). Observe how these operations affect the thickness of objects and the size of small gaps.
- **Assignment 16: Morphological Opening and Closing**
  Use **Opening** (erosion followed by dilation) to remove small noise objects and **Closing** (dilation followed by erosion) to fill holes within foreground objects.

Category 6: Feature & Shape Analysis

- **Assignment 17: Line Detection via Hough Transform**
  Transform an edge-detected image into the Hough Parameter Space (ρ, θ) to identify and highlight the most prominent linear structures in a scene.
- **Assignment 18: Harris Corner Detection**
  Implement the Harris Corner Detection algorithm to identify key points of high local variance in all directions. Mark these corners on the original image.
- **Assignment 19: Boundary Representation and Chain Codes**
  Extract the outer boundary of a segmented object and represent it using **Freeman Chain Codes** (4-connectivity or 8-connectivity). Discuss the benefits of this representation for shape storage.

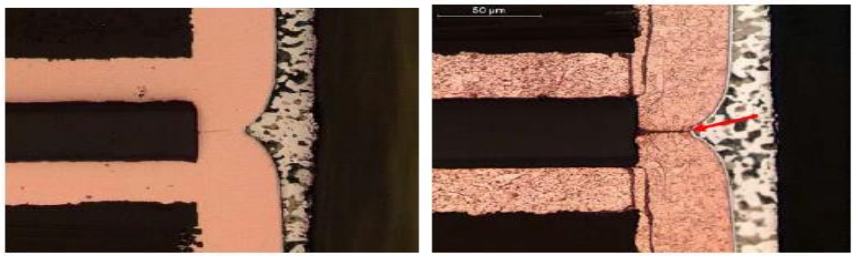
Category 7: Compression & Coding

- **Assignment 20: Image Compression and Quality Metrics**
  Implement a basic **Run Length Encoding (RLE)** scheme for binary images. Calculate the **Compression Ratio** and evaluate lossy compression performance using **Peak Signal-to-Noise Ratio (PSNR)**.

Syllabus
Home Assignments

List of Design based Home Assignments:

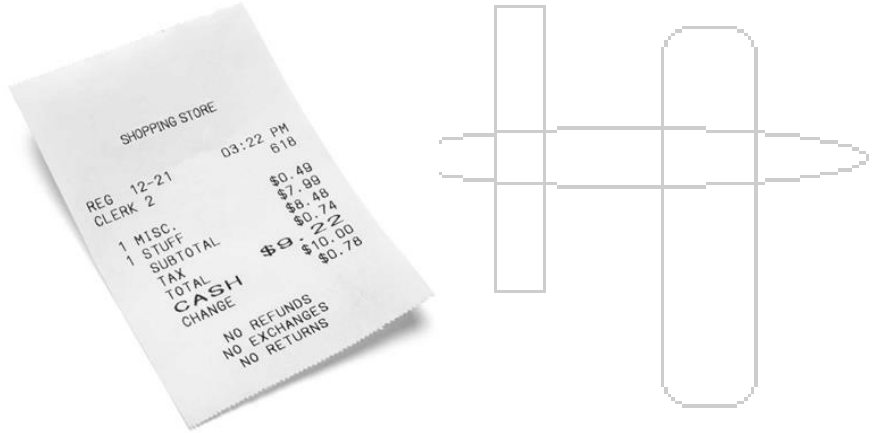
1.Design an algorithm to identify fault in a “PCB inspection system” as shown below



2.Design an algorithm to perform segmentation of the image below to extract the mango from its background.

3.Design an algorithm to get from image 1 to image 2.

4.Design an algorithm to recognize character “0” in the image below.



5.Design an algorithm to compress a 300x300 pixel image with horizontal black to white gradient as shown below.

List of Case Study based Home Assignments:

1. Cam-scanner: Document scanning app
2. Tesseract OCR library
3. Instagram filters
4. OpenCV
5. Google Street View

List of Blog based Home Assignments:

1. Image processing on Embedded platforms
2. Face recognition system security analysis for authentication
3. Image processing in MSME for effective automation
4. H.264 codec for image streaming
5. Role of mathematics in image processing

List of Survey based Home Assignments:

1. Image quality metrics
2. Vision based self-driving car safety
3. Compression techniques & codecs
4. State of the art applications such as AR/ XR
5. Human recognition in social networking apps like Facebook

CO-PO Mapping

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2		1	-	-	-	-	-	-	-	2	2	2	3	3
CO2	3	2	2	1	3	-	-	2	-	-	-	2	2	2	3	3
CO3	2	2	3	2	3	-	-	-	1	-	-	2	2	2	3	3
CO4	3	3	3	2	3	2	-	-	-	-	-	2	2	2	3	3
CO5	3	2		2	3	2	-	-	1	1	2	2	2	2	3	3
CO6	3	2		2	3	2	2	2		1	2	2	2	2	3	3
Average	**2.8**	**2.16**	**2.7**	**1.88**	**3**	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**2.0**	**2.0**	**2.0**	**2.0**	3.0	3.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L2, CO2 – L2, CO3 – L4, CO4 – L3, CO5 – L5 and CO6 – L3

Future Courses Mapping:

1. Augmented Reality
2. Multimedia Processing

Job Mapping:

1. Augmented Reality Experience Designer
2. Automation Engineer
3. Embedded Software Developer
4. Image Processing Expert

Text Books:

1. *Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.*
2. *Anil K. Jain, "Fundamental of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.*

Reference Books:

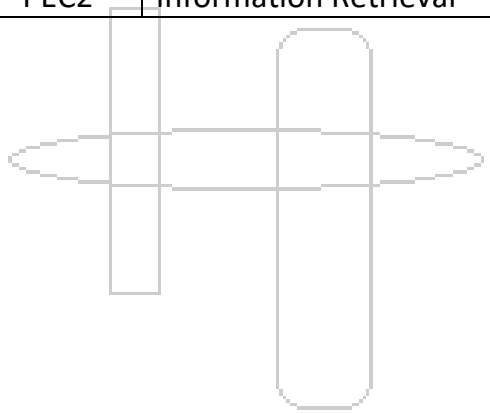
1.*Pratt, "Digital Image Processing," Wiley Publication, 3rdEdition, ISBN 0-471- 37407-5.*
2. *K.R. Castleman, "Digital Image Processing,"3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.*

For MOOCs and other learning Resources

1. https://nptel.ac.in/courses/117/105/117105135/
2. https://nptel.ac.in/courses/106/105/106105032/
3. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/

Program Elective Courses 2 (PEC2)

Course	Course Code
PEC2	MLOPs
PEC2	DevOps
PEC2	Natural Language Processing
PEC2	Information Retrieval



Syllabus Template **FF No. : 654**

ABCXXX: Machine Learning Operations (MLOps)

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 5 Hrs / Week

Course Link: https://www.coursera.org/specializations/machine-learning-operations-mlops

Course Relevance:

MLOps (Machine Learning Operations) is an engineering discipline that aims to unify machine learning system development and machine learning system operations. Coursera's MLOps catalogue teaches you how to streamline and regulate the process of deploying, testing, and improving machine learning models in production. You'll learn about essential elements of MLOps such as data and model versioning, model testing, monitoring, and validation, as well as robust strategies for deploying and maintaining ML models. By the end of your learning journey, you will be able to effectively manage the ML lifecycle, understand the role of automation in MLOps, and leverage best practices to bring data science and IT operations together. After completing this course, students will be able to:

1. Understand the fundamentals of MLOps, including its relationship with Machine Learning and DevOps.
2. Design and implement end-to-end ML pipelines, including data ingestion, preprocessing, model training, and evaluation.Build automated ML pipelines with GitHub Actions, serve models via FastAPI, and implement CI/CD workflows with Docker
3. Apply version control techniques for datasets, code, and models using tools like Git and DVC.
4. Deploy machine learning models into production using containerization and orchestration tools such as Docker and Kubernetes.
5. Monitor, maintain, and retrain ML models by identifying issues like data drift and performance degradation.
6. Implement CI/CD pipelines for ML workflows using tools like Jenkins or GitHub Actions.

Syllabus: It has 6 modules:

10. DevOps for Machine Learning
11. Build CI/CD pipelines with GitHub Actions to automate ML testing, training, and deployment workflows
12. Cloud Platforms for ML
13. Deploy ML models using AWS SageMaker endpoints, Azure Functions, and Google Cloud Vertex AI
14. Model Serving Systems
15. Build optimized Docker images and multi-container ML apps using Docker Compose and multi-stage builds

Syllabus Template **FF No. : 654**

ABCXXX: DevOps

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 5 Hrs / Week

Course Link: https://www.coursera.org/professional-certificates/ibm-applied-devops-engineering

Course Relevance:

DevOps Engineering is a modern approach to software development that integrates development (Dev) and operations (Ops) to improve collaboration, efficiency, and speed in delivering software applications.

Traditionally, development and operations teams worked separately—developers focused on writing code, while operations teams handled deployment and maintenance. This separation often led to delays, miscommunication, and system failures. DevOps addresses these challenges by bringing both teams together into a unified workflow.

After completing this course, students will be able to:

5. Understand the principles, culture, and lifecycle of DevOps, and its role in modern software development. JavaScript, React and Bootstrap
6. Design and implement Continuous Integration and Continuous Deployment (CI/CD) pipelines using tools such as Jenkins and GitHub Actions.
7. Develop and deploy applications using containerization technologies like Docker.
8. Implement Infrastructure as Code (IaC) and configuration management for scalable and repeatable deployments.
9. Monitor, log, and troubleshoot applications to ensure performance, reliability, and availability.
10. Apply security practices (DevSecOps) and ensure compliance in DevOps pipelines.

Syllabus: It has 9 modules:

16. Introduction to DevOps
17. Introduction to Agile Development and Scrum
18. Introduction to Containers w/ Docker, Kubernetes & OpenShift
19. Application Development using Microservices and Serverless
20. Introduction to Test and Behavior Driven Development
21. Continuous Integration and Continuous Delivery (CI/CD)
22. Application Security for Developers and DevOps Professionals
23. Monitoring and Observability for Development and DevOps
24. DevOps Capstone Project

Syllabus Template **FF No. : 654**

ABCXXX: Natural Language Processing

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 5 Hrs / Week

Course Link: https://www.coursera.org/specializations/natural-language-processing

Course Relevance:

Natural Language Processing (NLP) is a branch of Artificial Intelligence that enables computers to understand, interpret, and generate human language.It combines techniques from Machine Learning, linguistics, and computer science.NLP allows machines to process text and speech in a meaningful way.It is used in applications such as chatbots, translation systems, and voice assistants.Common NLP tasks include text classification, sentiment analysis, and language translation.It involves preprocessing steps like tokenization, stemming, and removing stop words.Advanced NLP uses deep learning models such as neural networks and transformers.Popular tools and libraries include TensorFlow and PyTorch.

After completing this course, students will be able to:

5. Understand the fundamentals of Natural Language Processing and its relationship with Artificial Intelligence and Machine Learning.
6. Apply text preprocessing techniques such as tokenization, stemming, lemmatization, and stop-word removal.
7. Implement basic NLP tasks including text classification, sentiment analysis, and information extraction.
8. Develop NLP applications using libraries like NLTK, spaCy, and TensorFlow.s
9. Apply deep learning techniques such as RNNs, LSTMs, and transformers for advanced NLP tasks.
10. Design and implement real-world applications such as chatbots, machine translation, and text summarization systems.

Syllabus: It has 4 modules:

10. Natural Language Processing with Classification and Vector Spaces
11. Natural Language Processing with Probabilistic Models
12. Natural Language Processing with Sequence Models
13. Natural Language Processing with Attention Models

ABCXXX: Information Retrieval

Credits: 3 **Learning Scheme: Coursera**

Duration: 4 Months - 5 Hrs / Week

Course Link: https://www.coursera.org/learn/text-retrieval

Course Relevance:

Information Retrieval (IR) is a field of Computer Science that focuses on retrieving relevant information from large datasets.It is designed to help users find useful information based on their queries.IR systems are commonly used in search engines like Google.Unlike databases, IR systems retrieve approximate matches rather than exact results.It deals mainly with unstructured data such as text documents and web pages.The core components of IR include indexing, query processing, and ranking.IR uses algorithms to determine the relevance of documents to a query.It is closely related to Natural Language Processing and machine learning

After completing this course, students will be able to:

5. Understand the fundamentals of Information Retrieval and its role in Computer Science.
6. Apply indexing techniques and data structures to efficiently organize and retrieve information.
7. Implement retrieval models such as Boolean, vector space, and probabilistic models.
8. Evaluate the performance of IR systems using metrics like precision, recall, and F-measure.
9. Apply IR techniques in real-world applications such as search engines and recommendation systems.
10. Understand challenges such as scalability, ambiguity, and handling unstructured data in IR systems

Syllabus: It has 6 modules:

9. Natural language processing techniques.
10. Basic idea of the vector space mode.
11. Evaluation of an information retrieval system
12. Probabilistic retrieval models and statistical language models
13. Feedback techniques in information retrieval,
14. Recommender systems - content-based recommendation/filtering and collaborative filtering.

Open Elective Courses (OE)

Course Code	Course Name
	Fintech
	Digital marketing
	Entrepreneurship
	Management

Syllabus Template **FF No. : 654**

MD3201: FinTech

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

The subject is highly relevant for engineering students as it provides knowledge of how technology is transforming the financial sector through digital banking, payments, data analytics, artificial intelligence, blockchain, and automation. It helps students understand financial systems, market operations, risk management, and the role of innovation in improving financial services. The subject also introduces important legal, regulatory, and cybersecurity considerations that are essential for developing secure and compliant financial solutions. By combining engineering, finance, and business concepts, FinTech equips students with practical skills that support careers in software development, financial technology, digital banking, cybersecurity, data analytics, and entrepreneurship. This interdisciplinary knowledge prepares graduates to contribute effectively to the growing digital financial ecosystem. In this course students will:

1. **Understand** the role of emerging technologies such as AI, blockchain, and digital payments in transforming financial services.
2. **Analyze** financial markets, instruments, and risk management concepts relevant to fintech applications.
3. **Apply** engineering and data-driven approaches to design and evaluate innovative financial technology solutions.
4. **Assess** regulatory, legal, security, and compliance requirements associated with digital financial systems.
5. **Develop** problem-solving and interdisciplinary skills for careers in fintech, digital banking, financial analytics, and technology-driven financial services.

Syllabus: This subject contains following courses

Innovations in Investment Technology: Artificial Intelligence	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/learn/invest-tech?source=search
AI in Financial Services: Foundations through future trends Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/ai-financial-services-foundations-future-trends?source=search
Innovations in Investment Technology: Artificial Intelligence	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/fintech-startups-emerging-markets?source=search
Blockchain Revolution in Financial Services Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-

	m0ein/specializations/blockchain-financial-services?source=search
Supply Chain Finance and Blockchain Technology Specialization	https://www.coursera.org/programs/faculty-development-program-b-2025-26-sem-i-m0ein/specializations/supply-chain-finance-and-blockchain-technology?source=search

Syllabus Template **FF No. : 654**

ABCXXX: Digital Marketing

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

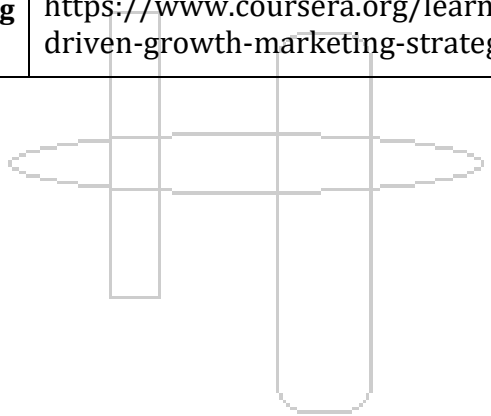
Digital Marketing provides knowledge and skills required to promote products, services, and innovations in the digital era. The subject introduces students to key concepts such as search engine optimization (SEO), social media marketing, content marketing, email marketing, web analytics, and online advertising. Engineering graduates often work on technology-based products and solutions, and understanding digital marketing helps them effectively communicate value to customers and stakeholders. The course also develops analytical skills through the use of data-driven marketing strategies, consumer behavior analysis, and performance measurement tools. Furthermore, it enhances students' understanding of emerging technologies, digital platforms, and business models that drive modern organizations. By combining technical expertise with marketing knowledge, students become better equipped for careers in technology management, entrepreneurship, product development, e-commerce, and digital business. The subject encourages creativity, strategic thinking, and customer-centric problem-solving, enabling engineers to contribute effectively to the growth and competitiveness of organizations in a digitally connected world. In this course students will:

1. Understand the fundamental concepts, tools, and techniques used in digital marketing and online promotion.
2. Apply digital marketing strategies including SEO, social media marketing, content marketing, and email marketing to real-world scenarios.
3. Analyze consumer behavior and marketing campaign performance using web analytics and data-driven decision-making techniques.
4. Develop effective digital marketing plans to enhance brand visibility, customer engagement, and business growth.
5. Evaluate emerging digital technologies and platforms to support innovation, entrepreneurship, and technology-based business solutions.

Syllabus: This subject contains following courses

Full Digital Marketing Specialization	https://www.coursera.org/specializations/digital-marketing-certification-course
Meta Social Media Marketing Professional Certificate	https://www.coursera.org/professional-certificates/facebook-social-media-marketing

Strategic SEO for Digital Marketing Success Specialization	https://www.coursera.org/specializations/digitalmarketingsuccess-strategic-seo
AI-Powered Modern Marketing Specialization	https://www.coursera.org/specializations/ai-powered-modern-marketing
Master AI for Digital Marketing & Content Creation	https://www.coursera.org/learn/ai-digital-marketing-content-creation
The Digital Marketing Revolution	https://www.coursera.org/learn/digital-marketing-revolution
Artificial Intelligence in Marketing	https://www.coursera.org/learn/uva-darden-artificial-intelligence-marketing
AI Marketing Mastery Specialization	https://www.coursera.org/specializations/ai-marketing-mastery
Generative AI in Marketing Specialization	https://www.coursera.org/specializations/generative-ai-in-marketing
Grow with AI: Your AI-driven Growth Marketing strategy	https://www.coursera.org/learn/grow-with-ai-your-ai-driven-growth-marketing-strategy



Syllabus Template **FF No. : 654**

ABCXXX: Entrepreneurship

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

Entrepreneurship is a highly valuable subject for engineering students as it develops the mindset and skills required to transform innovative ideas into successful business ventures. The subject introduces students to key concepts such as opportunity identification, business model development, innovation management, startup creation, market analysis, financial planning, and leadership. Engineering students often possess strong technical knowledge, and entrepreneurship helps them understand how to commercialize technologies and create solutions that address real-world problems. The course encourages creativity, critical thinking, risk assessment, decision-making, and problem-solving abilities that are essential for launching and managing new ventures. It also provides insights into funding strategies, customer needs, product development, and business growth. By integrating technical expertise with entrepreneurial skills, students become better prepared for careers as startup founders, innovators, project leaders, and business professionals. The subject fosters self-reliance, adaptability, and a proactive approach to identifying opportunities in a competitive global market, enabling engineering graduates to contribute to economic development and technological advancement. In this course students will:

1. Identify entrepreneurial opportunities and evaluate the feasibility of innovative business ideas.
2. Develop business models and startup strategies for technology-based products and services.
3. Apply problem-solving, leadership, and decision-making skills in entrepreneurial and organizational contexts.
4. Analyze market needs, customer requirements, and financial aspects for successful venture creation and growth.
5. Demonstrate an entrepreneurial mindset by managing risks, fostering innovation, and pursuing sustainable business opportunities.

Syllabus: This subject contains following courses

AI Innovation & Entrepreneurship Specialization	https://www.coursera.org/specializations/ai-innovation-and-entrepreneurship
Information Technology Entrepreneurship & Startup Specialization	https://www.coursera.org/specializations/it-entrepreneurship-startup
Technology Entrepreneurship Specialization	https://www.coursera.org/specializations/technology-entrepreneurship
Social Impact Strategy: Tools for Entrepreneurs and Innovators	https://www.coursera.org/learn/social-impact

Entrepreneurship: Launching an Innovative Business Specialization	https://www.coursera.org/specializations/business-entrepreneurship
Startup Entrepreneurship Specialization	https://www.coursera.org/specializations/startup-entrepreneurship
Innovation: From Creativity to Entrepreneurship Specialization	https://www.coursera.org/specializations/innovation-creativity-entrepreneurship
Social Entrepreneurship Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/social-entrepreneurship-cbs?source=search

Syllabus Template **FF No. : 654**

ABCXXX: Management

Credits: 4 **Learning Scheme: Coursera**
Duration: 4 Months - 10 Hrs / Week **Semester: II**
Course Relevance:

Management subject provides the knowledge and skills required to effectively plan, organize, lead, and control resources in professional and organizational settings. While engineering education develops strong technical expertise, management studies help students understand how to manage projects, teams, operations, and business processes efficiently. The subject introduces key concepts such as strategic planning, organizational behavior, leadership, decision-making, human resource management, quality management, and project management. It enables students to develop communication, teamwork, problem-solving, and analytical skills that are essential for successful careers in industry and business. Management knowledge also helps engineers understand organizational goals, customer requirements, resource utilization, and productivity improvement. Furthermore, it prepares students to take on leadership roles, manage multidisciplinary teams, and contribute to organizational growth and innovation. By combining technical competence with managerial capabilities, engineering graduates become better equipped to handle complex challenges, improve operational performance, and make informed decisions in dynamic and competitive environments. In this course students will:

1. Understand fundamental management principles, organizational structures, and business functions relevant to engineering practice.
2. Apply planning, organizing, leadership, and decision-making skills to manage projects and teams effectively.
3. Analyze organizational challenges and develop strategies for improving productivity, quality, and operational efficiency.
4. Demonstrate effective communication, teamwork, and leadership skills in multidisciplinary professional environments.
5. Evaluate managerial and strategic decisions considering organizational goals, resource utilization, and sustainable growth.

Syllabus: This subject contains following courses

Investment Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/investment-management?source=search
Strategic Leadership and Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/strategic-leadership?source=search

Financial Management Specialization	https://www.coursera.org/programs/fy-toppers-vit-b-2025-26-sem-i-q0138/specializations/financial-management?source=search

Multi-Disciplinary Minor Courses (MDM)

Course Code	Course Name
	Mathematics for Machine Learning
	Operational Research
	GenAI
	Randomized and Approximation Algorithms
	Business Intelligence
	Quantum Computing

Syllabus Template **FF No. : 654**

CS***: Mathematics for Machine Learning

Credits: 3 **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hour/Week

Course Prerequisites:
Prior exposure to matrices, analytical geometry, calculus, basic probability (typically taught at junior college level) is recommended but not mandatory. Familiarity with mathematical notation, basic set theory, and elementary logic is expected.

Course Objectives:
1. To establish a rigorous mathematical foundation in abstract vector spaces, linear transformations, and orthogonality to model multi-dimensional structures.
2. To explore spectral decompositions (Eigen, SVD) of matrices
3. To utilize multivariable calculus to solve optimization problems using gradient descent
4. To develop understanding towards linear regression, principal component analysis and its geometric interpretation
5. To introduce students to formal probability theory and distributions
6. To develop understanding towards mathematical foundations of Markov chains and Hidden Markov Model and study fundamental algorithmic problems governing HMMs

Course Outcomes:
The student will be able to –
1. Apply principles of linear algebra and analytical geometry to formalize the theoretical structures underlying multi-dimensional spaces.
2. Analyze the spectral properties of matrices to mathematically justify rank approximations and data transformations.
3. Calculate multivariate gradients and implementing algorithms like gradient descent and backpropagation to minimize objective functions.
4. Develop understanding towards the mathematical derivations and geometric interpretations behind linear regression, Principal Component Analysis (PCA).
5. Solve problems based on basic probability theory and distributions

Understand mathematical foundations of Markov Chains and Hidden Markov Models.

Course Relevance:
While modern machine learning frameworks often abstract away underlying mathematical structures, true innovation in the field requires a profound mastery of its mathematical foundations. This course aims at building mathematical foundations required for machine learning. By focusing on mathematical proofs, geometric interpretations, and analytical derivations of concepts from linear

algebra, analytical geometry, vector calculus, probability theory, etc students will develop the theoretical maturity required to not only implement machine learning algorithms, but to fundamentally understand and justify them. Ultimately, this course provides a bridge between pure abstract mathematics and its practical applications in machine learning/ data science.

Syllabus

Theory

Section 1: Topics/Contents

Unit 1: **Linear Algebra and Analytical Geometry** (5 Hours)

scalars, vectors, matrices, matrix multiplication, system of linear equations, elementary operation on matrices, row reduced echelon form, rank of matrix, inverse of matrix, vector spaces, subspaces, linear dependence and span of vectors, basis and dimension of vector space, null space, rank-nullity theorem, linear transformations, determinant function.

norms, inner products, lengths and distances, angles and orthogonality, orthonormal basis, orthogonal complement, orthogonal projections.

Unit 2: **Matrix Decomposition** (4 Hours)

Eigen values and Eigen vectors, Eigen decomposition and diagonalization, singular value decomposition, best rank-k approximation, example: singular vectors and ranking documents,

Unit 3: **Optimization and Gradient Descent** (5 Hours)

Differentiation of univariate functions, partial differentiation and gradients, gradients of vector-valued functions, gradients of matrices, useful identities for computing gradients, backpropagation and automatic differentiation, higher-order derivatives, linearization and multivariate Taylor series, optimization using gradient descent.

Section 2: Topics/Contents

Unit 4: **Regression and Dimensionality Reduction** (5 Hours)

Introduction to regression, line fitting, least square method, generalization to higher dimensions, overfitting, regularization, Bayesian linear regression.

Introduction to Principal component analysis, dimension reduction and its importance, curse of dimensionality, direction of maximum variance, geometric interpretation, covariance matrix, eigen vectors of covariance matrix and principal components, some real-world examples.

Unit 5: **Probability and distributions** (5 Hours)

basic probability theory: basic counting, definition of probability, construction of a probability space, discrete and continuous probabilities, examples, independence of events, chain rule and conditioned probability, sum rule, product rule and Bayes' theorem.

random variables, expectation, variance of random variable, probability distributions, some common probability distributions (Uniform, Bernaulli, Binomial, Geometric, Gaussian),

Unit 6: **Markov Chains and Hidden Markov Models** (4 Hours)

Sequential data, deterministic vs stochastic systems, Markov property, Discrete time Markov chains, state transition diagram, classifying states of Markov chains (transient vs recurrent states), Google's page-rank algorithm (to be discussed in tutorials)

Introduction to Hidden Markov models, observation space, transition and emission probabilities, examples, fundamental algorithmic problems governing HMMs (evaluation, decoding, learning), Viterbi algorithm (to be discussed in tutorials)

Tutorials

List of Tutorials

1. Problem solving based on vector spaces, subspaces, linear independence, and basis.

   Example Tasks: Prove whether a given set of vectors forms a subspace. Find the basis, dimension, and null space (kernel) of a given linear transformation matrix.

2. Problem solving based on Inner products, norms, orthogonal complements, and the Gram-Schmidt process.

   Example Tasks: Compute the orthogonal projection of a vector onto a subspace. Prove the Pythagorean theorem in an inner product space. Find an orthonormal basis for a given span.

3. Problem solving based on eigenvalues, eigenvectors, and Eigen decompositions.

   Example Tasks: Find Eigen values and Eigen vectors for a given matrix, determine if a given square matrix is diagonalizable.

4. Deriving Singular Value Decomposition and understanding the Eckart-Young-Mirsky theorem for low-rank approximations.

   Example Tasks: Compute the SVD of a small non-square matrix. Calculate the best rank k approximation of a matrix and quantify the approximation error.

5. Problem solving based on Partial derivatives, Jacobian matrices, Hessian matrices, and the multivariate Taylor series.
   Example Tasks: Compute the gradient and Hessian of a multi-variable quadratic function. Manually trace a single step of backpropagation using the chain rule on a simple 2-layer computational graph.
6. Problem solving based on Linear Regression. Example Tasks: Mathematically derive the closed-form solution for ordinary least squares.
7. Problem solving based on Principle component analysis. Example Tasks: Compute the covariance matrix for a small 2D dataset. Formulate PCA as a constrained optimization problem (maximizing variance subject to unit vector constraints) and solve it using Lagrange multipliers to show that the solution is an eigenvector.
8. Problem solving based on basic probability theory.
9. Understanding Google's Page-rank algorithm.

10. Problem solving based on state transition matrices, stationary distributions, and dynamic programming in HMMs.
11. Understanding Viterbi algorithm. Hand-trace the Viterbi algorithm on a highly simplified HMM (e.g., 2 hidden states, 2 observation types) to find the most likely sequence of hidden states.

CO-PO Mapping

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	3	1	1	-	-	-	-	-	-	-	1	2	1	-	1
2	3	3	2	2	1	-	-	-	-	-	-	1	3	2	1	2
3	3	3	2	2	2	-	-	-	-	-	-	1	3	2	1	2
4	3	3	2	2	2	-	-	-	-	-	-	1	3	2	2	2
5	3	3	1	1	1	-	-	-	-	-	-	1	2	1	-	1
6	3	3	2	2	2	-	-	-	-	-	-	1	3	2	2	2
Avg	**3**	**3**	**1.67**	**1.67**	**1.6**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2.67**	**1.67**	**1.5**	**1.67**

Future Courses Mapping: Machine learning, Deep learning and Neural networks, Natural language processing, Probabilistic graphical models, reinforcement learning, advanced convex optimization.

Job Mapping: The rigorous mathematical and theoretical frameworks developed in this course are directly relevant to domains such as Advanced Machine learning, Deep learning, Quantitative Finance, Statistical Data Science, and Operations Research. Typical job titles include: machine learning researcher, data scientist, ML algorithm developer

Text Books:

[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 Learning from Data*. Wellesley, MA, USA: Wellesley-Cambridge Press, 2019.
[3] K. L. Chung, *Elementary Probability Theory with Stochastic Processes*. New York, NY, USA: Springer, 1974.
[4] R. Motwani and P. Raghavan, *Randomized Algorithms*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1995.
[5] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[6] S. Kumaresan, *Linear Algebra: A Geometric Approach*. New Delhi, India: Prentice-Hall of India, 2000.

Reference Books: (As per IEEE format)

[7] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of Data Science*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[8] K. Hoffman and R. Kunze, *Linear Algebra*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1971.

CS***: Operational Research

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

Tutorial: 1 Hour/Week

Course Prerequisites:

Mathematics, Probability and Statistics, Linear Algebra, Basic Programming

Course Objectives:

1. Understand the fundamentals and applications of Linear Programming (LP) in decision-making and optimization.
2. Apply the Simplex algorithm to obtain optimal solutions for large-scale Linear Programming problems.
3. Understand the concepts and theoretical foundations of duality in Linear Programming.
4. Analyze and solve variants of the transportation problem
5. Apply network optimization techniques to solve shortest path, minimum spanning tree, maximum flow problems
6. Apply appropriate algorithms and techniques to solve Integer Linear Programming problems.

Course Outcomes:

1. Formulate Linear Programming models by identifying decision variables, objective functions, and constraints from real-world scenarios.
2. Solve Linear Programming problems using the Simplex Method and identify optimal solutions.
3. Perform post-optimal (post-optimality) analysis to assess the effects of changes in objective function coefficients, constraints, and resource availability.
4. Solve transportation problems using appropriate methods to obtain optimal transportation schedules.
5. Solve network optimization problems, including shortest path, minimum spanning tree, maximum flow,
6. Solve Integer Linear Programming problems using appropriate techniques such as Branch and Bound, Cutting Plane

Course Relevance:

For engineering students, Operations Research provides a systematic approach to analyzing real-world problems in manufacturing, production planning, supply chain management, transportation, project scheduling, network design, and resource allocation. It introduces optimization techniques such as Linear Programming, Integer Programming, Transportation Models, Network Models, and Sensitivity Analysis, which are widely used in modern industries.

Syllabus
Theory

Section 1: Topics/Contents

Unit 1: Introduction and Modelling with Linear Programming (4 Hours)

Operations Research Models, Solving the OR Model, Queuing and Simulation Models, Art of Modelling

Two-variable LP Model, Graphical LP Solution, Linear Programming Applications

Unit 2: The Simplex Method and Sensitivity Analysis (5 Hours)

LP Model in equation form, Transition from graphical to algebraic solution, artificial starting solution, Special cases in simplex method, Sensitivity analysis

Unit 3: Duality and Post-optimal analysis (5 Hours)

Definition of the Dual Problems, Primal-Duality relationship, Economic interpretational of duality, Additional simplex algorithms: Dual simplex algorithm, Generalized simplex algorithm

Section 2: Topics/Contents

Unit 4: Transportation problem and its variants (5 Hours)

Definition of transportation model, Non-traditional transportation models, The transportation algorithm, The assignment model,

Unit 5: Network Model (4 Hours)

Scope and Definition of Network Models, Minimal Spanning tree algorithm, Shortest route problem, Maximal flow model,

Unit 6: Integer Linear Programming (5 Hours)

Illustrative applications: Capital budgeting, Set-covering problem, Fixed-charge problem, Either-or, if-then constraints

Modelling NP optimization problems as integer linear programs. Integer Programming Algorithms: Branch and Bound algorithm, Cutting-Plane algorithm

Tutorials

List of Tutorials

1. Solve examples of Linear Programming model.(2)
2. Solve examples of Simplex Method (2)
3. Solve examples of Transportation problem (2)
4. Examples of Maximum flow model (2)
5. Examples of integer linear programming

CO-PO Mapping

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	3	2	1	1	-	-	-	-	-	-	1	2	1		
2	3	3	2	2	2	-	-	-	-	-	-	1	2	1		
3	3	3	2	3	2	-	-	-	-	-	1	2	2	1		
4	3	3	3	2	2	-	-	-	-	-	2	1	2	1		
5	3	3	3	2	2	-	-	-	-	-	2	1	2	1		
6	3	3	2	2	2	-	-	-	-	-	1	2	2	1		
Avg	**3**	**3**	**2.33**	**2**	**1.83**	**-**	**-**	**-**	**-**	**-**	**1.5**	**1.33**	**2**	**1**		

Future Courses Mapping:

Advanced optimization techniques

Job Mapping:

Operations Research Analyst, Business Analyst, Data Analyst

Text Books:

Hamdy A. Taha, Operations Research An Introduction, TENTH EDITION, Pearson

Reference Books: (As per IEEE format)

Hiller/Liberma, Introduction to operations research, Seventh Edition

Syllabus Template **FF No. : 654**

CSM: GenAI

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

Tutorial: 1 Hour/Week

Course Prerequisites: Discrete mathematics (Probability), Data structures, Artificial Intelligence

Course Objectives:

1. To learn the fundamentals of Generative AI models and the frameworks used to build them.
2. To explore the fundamentals of Generative AI models, including GANs, VAEs and their applications.
3. To understand the concepts of transformer models and attention mechanisms used in Generative AI.
4. To provide fundamental knowledge of Large Language Models (LLMs), their fine-tuning, and real-world applications
5. **To implement prompt engineering techniques and their use with Large Language Models (LLMs).**
6. To apply theoretical understanding to hands-on projects, solving problems using Generative AI models.

Course Outcomes:

The student will be able to –

1. Understand the fundamentals of Generative AI models, their architectures, and the frameworks used for developing Generative AI applications.
2. Implement and evaluate Generative AI models such as GANs and VAEs for solving domain-specific problems.
3. Analyze transformer architectures and attention mechanisms for sequence modelling and generative tasks.
4. Apply Large Language Models (LLMs), including fine-tuning techniques, to develop AI solutions for real-world applications.
5. Design effective prompts using prompt engineering techniques such as zero-shot, few-shot, and chain-of-thought prompting to improve LLM performance.
6. Develop and demonstrate Generative AI applications by integrating learned concepts, frameworks, and models to solve practical problems.

Course Relevance: Generative AI is rapidly transforming the way intelligent systems are designed and deployed across various domains. This course provides students with essential knowledge of

modern Generative AI techniques, including transformer models, Large Language Models (LLMs), and prompt engineering. It emphasizes both theoretical concepts and practical implementation through projects using industry-relevant frameworks and tools. Students gain the skills required to develop AI-powered solutions for real-world applications. The course also prepares learners for advanced studies, research, and emerging career opportunities in Artificial Intelligence, Machine Learning, and Generative AI.

Syllabus
Theory

Section 1: Topics/Contents

Unit 1: Introduction (5 Hours)

Overview of generative models: Revision of concepts of neural networks, Importance of generative models in machine learning TensorFlow Basics: Introduction to TensorFlow and its computational graph, Tensors, operations, and variables PyTorch Basics: Introduction to PyTorch and its dynamic computation graph, Tensors, operations, and variables in PyTorch.

Unit 2: Generative AI Models (6 Hours)

Introduction to Generative AI Models: Variational Autoencoders (VAEs): Introduction to autoencoders, Variational inference, Implementing Variational Autoencoders, Applications of VAEs in generative modelling. Generative Adversarial Networks (GANs): Introduction to GANs, GAN architecture: generator and discriminator, Training GANs, Conditional GANs and their applications.

Unit 3:Transformer Models and Attention Mechanism (5 Hours)

Sequence Generation with Recurrent Neural Networks (RNNs): Introduction to sequence generation, Review of recurrent neural networks Training RNNs for sequence generation. Limitations of traditional sequential models, Introduction to transformer models, Attention mechanism in generative models, Implementing transformers for sequence generation.

Section 2: Topics/Contents

Unit 4: Large Language Models (5 Hours)

Overview of Large Language Models, Pretrained LLMs: Loading and using models from Hugging Face, Case study: text generation and completion. Fine-Tuning LLMs: Introduction to fine-tuning concepts, preparing datasets, Fine-tuning models Case study: implementation of LLMs in customer support, analysing challenges, solutions, and impact

Unit 5: Prompt Engineering (4 Hours)

Intro to Prompt Engineering, LLM with Prompt Engineering, role of prompts, Fine-tuning vs. Prompt Engineering, types of prompts and their uses, Zero-shot, few-shot, and chain-of-thought prompting, Using LangChain for prompt management

Unit 6: Generative AI Applications: (3 Hours)

Applications in Various Fields: Art and Creativity, Image and Video Generation, Text Generation, Music Composition, Healthcare, Finance. Real-world use cases and challenges in deploying generative AI models.

Tutorials

List of Tutorials

1. Build and train a basic neural network using TensorFlow for a classification task.
2. Build and train a basic neural network using PyTorch for a classification task.
3. Implement a simple Generative Adversarial Network for image generation using PyTorch
4. Implement a basic autoencoder for image reconstruction using TensorFlow or PyTorch.
5. Implement an RNN/LSTM-based model for sequence generation using either TensorFlow or PyTorch
6. Implement a Transformer model for a sequence-to-sequence task using PyTorch or TensorFlow
7. Design zero-shot, few-shot, chain-of-thought, and role-based prompts to improve LLM responses for different tasks.
8. Develop an end-to-end application such as an AI-powered chatbot, document summarizer, content generator, code assistant, or image caption generator using an LLM and prompt engineering techniques.

Course Project

List of Course Projects

1. AI-Powered Document Summarizer
2. AI Interview Preparation Assistant
3. Smart Email Generator
4. PDF Question-Answering Chatbot
5. Intelligent Resume Analyzer
6. Customer Support Chatbot
7. Quiz Generator from Study Material
8. AI Grammar and Writing Assistant
9. Image Caption Generator

CO-PO Mapping

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	1	1	2	-	-	-	-	-	-	2	3	2	1	1
2	3	3	2	3	3	-	-	-	1	1	-	2	3	3	2	2
3	3	3	2	2	3	-	-	-	-	1	-	2	3	3	2	2
4	3	3	3	2	3	-	-	1	1	1	1	2	3	3	3	2
5	2	2	3	2	3	1	-	1	1	2	-	3	2	3	3	2
6	3	3	3	3	3	1	-	1	2	2	2	3	3	3	3	3
Avg	**3**	**2.66**	**2.33**	**2.16**	**2.83**	**0.33**		**0.5**	**0.83**	**1.16**	**0.5**	**2.33**	**2.83**	**2.83**	**2.33**	**2**

CO Attainment levels:
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55
CO1 – L3, CO2– L4, CO3 – L4, CO4 – L4, CO5 – L3 and CO6 – L5

Future Courses Mapping: Advanced AI, MLOps/LLMOps

Job Mapping: The Generative AI course prepares students for roles such as Generative AI Engineer, AI/ML Engineer, Prompt Engineer, LLM Application Developer, NLP Engineer, Data Scientist, AI Research Associate.

Text Books:
[1] Jacob Emerson: “Ripples of Generative AI: How Generative AI Impacts, Informs and Transforms Our Lives”
[2] David Foster: “Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play” O”Reilly Publication
[3] Altaf Rehmani: “Generative AI for everyone: Understanding the essentials and applications of this breakthrough technology"

Reference Books:
[1] Alger Fraley: The Artificial Intelligence and Generative AI Bible: [5 in 1] The Most Updated and Complete Guide”
[2] Joseph Babcock, Raghav Bali: Generative AI with Python and TensorFlow 2:
[3] "Generative Adversarial Networks Cookbook: Over 100 recipes to build generative models using Python, TensorFlow, and Keras" by Josh Kalin.

For MOOCs and other learning Resources:
1. Generative AI with Large Language Models, Coursera Available at: Generative AI with Large Language Models | Coursera

2. Generative AI for Everyone, Coursera Available at: Generative AI for Everyone | Coursera
3. Generative AI: Introduction and Applications, Coursera Available at: Generative AI: Introduction and Applications | Coursera

Syllabus Template **FF No. : 654**

CS***: Randomized and Approximation Algorithms

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

Tutorial: 1 Hour/Week

Course Prerequisites: Discrete mathematics, Design and Analysis of Algorithms, Data structures, familiarity with basic probability

Course Objectives:

1. Apply discrete probability theory to mathematically model and evaluate the expected performance of randomized algorithms.
2. Utilize moment deviations and tail inequalities (Markov, Chebyshev, and Chernoff) to rigorously bound algorithmic runtimes and error probabilities.
3. Design and differentiate between Las Vegas and Monte Carlo algorithms and understand their theoretical placement within randomized complexity classes (RP, BPP, ZPP).
4. Understand application of algebraic techniques in randomized algorithms, including Schwartz-Zippel lemma and Isolation lemma.
5. Understand randomized number-theoretic algorithms, formally deriving the theoretical guarantees on algorithms like Miller-Rabin primality test, Solovay Strassen primality test.
6. Understand the mathematical foundations of Markov chains and random walks to analyze key structural properties like stationary distributions, hitting times, and rapid mixing.

Course Outcomes:

The student will be able to –

1. Analyze complexity and success probability of randomized algorithms using tools from basic probability theory.
2. Formulate rigorous mathematical bounds on algorithmic deviation and error probabilities using tail inequalities.
3. Understand various randomized complexity classes, probability amplification techniques.
4. Prove the correctness of randomized algebraic algorithms utilizing foundational tools like the Schwartz-Zippel lemma and Isolation lemma.
5. Assess randomized number-theoretic algorithms, formally deriving the theoretical guarantees on algorithms like Miller-Rabin primality test.
6. Understand mathematical foundations of Markov chains and random walks and their applications.

Course Relevance: As randomized algorithms often outperform the deterministic algorithms for the problems, randomness emerges as an essential mathematical tool for designing optimal algorithmic solutions. This course provides the rigorous probabilistic framework necessary to design and analyze randomized algorithms. By mastering advanced theoretical constructs such as concentration inequalities, algebraic techniques like identity testing, and the spectral properties of Markov chains, students will formally prove the computational advantages of randomized paradigms. Ultimately, this theoretical grounding equips students with the analytical maturity required to develop and analyze randomized algorithms for important problems.

Syllabus

Theory

Section 1: Topics/Contents

Unit 1: **Basic Probability theory** (5 Hours)

Introduction to randomization in computation and some simple randomized algorithms. Basic discrete probability theory: basic counting, definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes' rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, Indicator random variables and their role in algorithm analysis.

Unit 2: **Tail inequalities and applications** (4 Hours)

Moments and deviation, Tail inequalities (Markov, Chebyshev and Chernoff's) and their applications, weak law of large numbers, sharp threshold for coupon collector problem.

Unit 3: **Randomized Algorithms and Randomized Complexity Classes.** (5 Hours)

Las Vegas and Monte-Carlo algorithms (with examples: randomized quick sort, Karger's min-cut algorithm), Abundance of witnesses, Basic complexity classes P, NP, RP, Co-RP, ZPP, BPP and their interrelations, probability amplification in RP and BPP.

Section 2: Topics/Contents

Unit 4: **Algebraic Techniques** (5 Hours)

Polynomial identity testing, Schwartz-Zippel lemma and applications (with examples verifying matrix multiplication, testing equality of strings, perfect matching problem for bipartite graphs), Mulmuley-Vazirani-Vazirani isolation lemma and application to matching problem.

Unit 5: **Number theoretic algorithms** (4 Hours)

Quadratic residues and nonresidues, computing non-residues and applications, Miller-Rabin primality test, Solovay-Strassen primality test, prime number theorem and generating large primes.

Unit 6: **Markov Chains and Random Walks** (5 Hours)

Markov chains: definition, representations, randomized algorithm for 2-SAT and 3-SAT, classifying states of Markov chains, Gambler's ruin, stationary distributions. Random walks on undirected graphs, cover time, hitting time, commute time, graph connectivity, electrical networks, introduction to expander graphs. Expanders and rapidly mixing random walks.

Tutorials

List of Tutorials

12. Problem solving based on basic discrete probability (topics: Bayes' rule, independence of events, union bounds, and linearity of expectation).
13. Assignment based on study of classical distributions including Bernoulli, Binomial, Geometric distributions.
14. Assignment based on probability amplification of algorithms
15. Assignment based on tail inequalities where the students will start with a basic coin-tossing scenario and apply all three inequalities to bound the probability of getting significantly more heads than expected, comparing the "tightness" of each bound.
16. Dry-running Karger's Min-Cut on a small example graph. Students will then calculate the exact probability of finding the minimum cut in that specific graph and discuss the need to repeat the algorithm.
17. Understanding Schwartz-Zipple lemma and isolation lemma through problem solving.
18. Hand-computing quadratic residues and non-residues modulo a small prime, followed by dry-running one iteration of the Miller-Rabin primality test on a small composite number to find a "witness" to its compositeness.
19. Constructing the Markov chain transition matrix for the randomized 2-SAT algorithm. Students will use the concept of a 1D random walk (Gambler's ruin) to calculate the expected time to reach a satisfying assignment.

CO-PO Mapping

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	3	1	1	1	-	-	-	-	-	-	1	3	-	-	1
2	3	3	1	2	1	-	-	-	-	-	-	1	3	-	-	2
3	3	2	1	1	-	-	-	-	-	-	-	1	3	1	-	1
4	3	3	2	2	-	-	-	-	-	-	-	1	3	-	1	2
5	3	3	2	2	-	-	-	-	-	-	-	1	3	-	1	2
6	3	3	2	2	2	-	-	-	-	-	-	1	3	1	2	2
Avg	**3**	**2.83**	**1.5**	**1.67**	**1.33**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**3**	**1**	**1.33**	**1.67**

Future Courses Mapping: Computational complexity theory, mathematical foundations of cryptography, quantum computing

Job Mapping: The course is relevant for mathematical research in the domain of algorithms, complexity theory, cryptography.

Text Books:

[1] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge University Press, 1995.

[2] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2017.

[3] K. L. Chung, *Elementary Probability Theory with Stochastic Processes*. New York, NY, USA: Springer, 1974.

Reference Books:

[4] N. Alon and J. H. Spencer, *The Probabilistic Method*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2016.

[5] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*, 2nd ed. Providence, RI, USA: American Mathematical Society, 2017.

[6] J. Hromkovič, *Design and Analysis of Randomized Algorithms: Introduction to Design Paradigms*. Berlin, Germany: Springer, 2005.

CS***: Business Intelligence

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

Tutorial: 1 Hour/Week

Course Prerequisites: DBMS & SQL fundamentals, Basic programming (Python/Java), Statistics basics (mean, probability), Data handling (Excel/CSV, preprocessing).

Course Objectives:

After completing this course, students will be able to:

1. Understand the fundamentals and architecture of Business Intelligence systems.
2. Explain data warehousing concepts, ETL processes, and data modeling techniques.
3. Apply OLAP operations for multidimensional data analysis.
4. Analyze and apply data mining techniques for business decision-making.
5. Design effective dashboards and visualizations using BI tools like Power BI and Tableau.
6. Explore advanced BI concepts including big data, cloud BI, and predictive analytics.

Course Relevance:

The Business Intelligence course equips students with essential skills to transform data into actionable insights, supporting data-driven decision-making in industries such as finance, healthcare, and retail. It aligns with current industry demands for analytics, visualization, and data-driven strategies.

Section 1: Topics/Contents

Unit I: Introduction to Business Intelligence (4 Lectures)

Definition and Importance of BI, Evolution from Data to Information to Knowledge, BI vs Data Science vs Data Analytics, Components of BI Architecture, Role of BI in organizations, Real-world BI applications.

Unit II: Data Warehousing Concepts (6 Lectures)

Introduction to Data Warehouse, OLTP vs OLAP, Data Warehouse Architecture, ETL Process (Extract Transform Load), Data Modeling (Star Schema Snowflake Schema), Data Marts and Metadata.

Unit III: OLAP and Multidimensional Data Analysis (5 Lectures)

OLAP Concepts and Types (ROLAP MOLAP HOLAP), Multidimensional Data Model, OLAP Operations (Slice Dice Drill-down Roll-up Pivot), Querying OLAP Cubes, OLAP Performance Issues.

Unit IV: Data Mining for BI (5 Lectures)

Introduction to Data Mining, BI and Data Mining Integration, Classification (Decision Trees), Clustering (K-Means), Association Rule Mining (Apriori), Applications and Challenges of Data Mining.

Unit V: Data Visualization & Reporting (4 Lectures)

Importance of Data Visualization, Principles of Effective Visualization, Types of Charts and Graphs, Dashboard Design and KPIs, Tools (Power BI Tableau).

Unit VI: Advanced BI Concepts (4 Lectures)

Big Data and BI, Real-time BI, Cloud BI, Predictive Analytics, AI in BI

Syllabus
Laboratory

List of Lab Assignments:

1. Analyze a real-world scenario to demonstrate how BI transforms data into actionable insights.
2. Design a Business Intelligence architecture showing data sources, ETL, warehouse, and reporting layers.
3. Perform data preprocessing on a dataset including cleaning, handling missing values, and normalization.
4. Implement an ETL pipeline to extract, transform, and load data into a database.
5. Design and implement a star or snowflake schema for a given application.
6. Perform OLAP operations (slice, dice, drill-down, roll-up) on a dataset.
7. Create and analyze a multidimensional data cube for business insights.
8. Build and evaluate a decision tree classification model on a dataset.
9. Apply K-Means clustering to identify patterns and visualize clusters.
10. Implement Apriori algorithm to generate association rules from transactional data.
11. Develop an interactive dashboard using Power BI or Tableau.
12. Build a predictive model to forecast outcomes and visualize insights.

List of Course Projects:

1. Excel-Based Sales Data Analysis Dashboard
2. Basic Data Cleaning and Visualization of Retail Dataset
3. SQL-Based Business Reporting System for Sales Data

4. ETL Pipeline for Integrating Multi-Source Business Data
5. Data Warehouse Design for Retail Management System
6. OLAP-Based Sales Analysis and Reporting System
7. Customer Segmentation using K-Means Clustering
8. Decision Tree-Based Customer Purchase Prediction System
9. Market Basket Analysis using Apriori Algorithm

10. Interactive BI Dashboard using Power BI or Tableau
11. Real-Time Business Analytics Dashboard using Streaming Data
12. End-to-End BI System with Predictive Analytics and AI Integration

Course Outcomes:

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

1. Describe BI concepts, architecture, and its role in decision-making. *(Understand)*
2. Implement data warehousing techniques including ETL and schema design. *(Apply)*
3. Perform OLAP operations for data analysis. *(Apply)*
4. Apply data mining algorithms (classification, clustering, association) to business problems. *(Analyze)*
5. Develop interactive dashboards and reports using BI tools. *(Create)*
6. Evaluate modern BI trends such as big data analytics and AI-driven BI systems. *(Evaluate)*

Books and E-Resources

For Reference Print Book –

1. E. Turban, R. Sharda, D. Delen; *Business Intelligence: Data Analysis and Decision Making*; 10th Edition; Pearson; 2018
2. P. Ponniah; *Data Warehousing Fundamentals*; 2nd Edition; Wiley; 2010
3. R. Kimball, M. Ross; *The Data Warehouse Toolkit*; 3rd Edition; Wiley; 2013
4. J. Han, M. Kamber, J. Pei; *Data Mining: Concepts and Techniques*; 3rd Edition; Elsevier; 2011

For Reference Electronic Book –

1. T. H. Davenport, J. G. Harris; *Competing on Analytics: The New Science of Winning*; 1st Edition; Harvard Business Review Press; 2007
2. C. N. Knaflic; *Storytelling with Data: A Data Visualization Guide for Business Professionals*; 1st Edition; Wiley; 2015
3. R. Elmasri, S. B. Navathe; *Fundamentals of Database Systems*; 7th Edition; Pearson; 2016

For MOOCs and other learning Resources

1. *Data Warehousing for Business Intelligence Specialization*; Coursera; Available at: Data Warehousing for Business Intelligence | Coursera
2. *Google Business Intelligence Professional Certificate*; Coursera; Available at: Google Business Intelligence Professional Certificate | Coursera
3. *Business Analytics & BI Courses*; edX; Available at: Online business analytics courses | edX

CO-PO Mapping

	Program Outcomes (PO)												PSO			
CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	-	-	-	-	-	-	-	2	-	2	3	2	1	2
CO2	3	3	2	2	3	-	-	-	-	-	-	2	3	2	2	3
CO3	2	3	2	2	2	-	-	-	-	-	-	2	2	3	1	3
CO4	2	3	2	3	2	-	-	-	-	-	-	2	2	3	1	3
CO5	2	2	3	2	3	-	-	-	2	2	-	2	2	2	3	3
CO6	2	3	-	2	2	1	1	1	-	-	1	2	2	3	2	3
Average	2.3	2.7	2.2	2.2	2.3	1.0	1.0	1.0	2.0	2.0	1.0	2.2	2.3	2.5	1.7	2.8

CO Attainment Levels:

Weights for attainment levels:

L1 – Easy – 0.75, L2 – Comfortable – 0.70, L3 – Medium – 0.65,

L4 – Somewhat Difficult – 0.60, L5 – Difficult – 0.55

CO1 – L3, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L4, CO6 – L5

Future Courses Mapping (Business Intelligence):

The Business Intelligence course provides a strong foundation for advanced courses such as Data Analytics, Big Data Analytics, Machine Learning, Data Science, Artificial Intelligence, and Cloud Computing. The concepts of data warehousing, OLAP, data mining, and visualization learned in BI enable students to pursue specialized domains like predictive analytics, deep learning, and AI-driven decision systems, as well as industry-oriented courses focusing on real-time analytics and business strategy.

Syllabus Template **FF No. : 654**

CS***: Quantum Computing

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

Tutorial: 1 Hour/Week

Course Objectives:

1. To introduce the fundamental principles of quantum computation, quantum information, and the structure of quantum programming.
2. To develop the ability to construct and manipulate quantum circuits using quantum gates, registers, and measurement operations.
3. To provide hands-on experience in implementing quantum algorithms using modern quantum programming frameworks such as Qiskit or Q#.
4. To enable students to analyze and evaluate quantum program execution through simulation and real quantum hardware while considering practical challenges such as noise and hardware constraints.
5. To equip students with the skills to design, develop, and deploy small-scale hybrid quantum-classical applications for solving computational problems.

Course Outcomes:

Students will be able to

1. Understand the principles of quantum computation and distinguish between classical and quantum computational models. (Bloom's Level: Understand)
2. CO2. Analyze the complexity and structure of quantum algorithms such as Deutsch–Jozsa, Simon's, and Grover's algorithms. (Bloom's Level: Analyze)
3. Construct quantum circuits implementing foundational algorithms like Quantum Fourier Transform and phase estimation. (Bloom's Level: Apply/Create)
4. Implement quantum algorithms using frameworks such as Qiskit or Cirq and simulate their execution. (Bloom's Level: Apply)
5. Evaluate the computational advantages and limitations of quantum algorithms over classical counterparts. (Bloom's Level: Evaluate)
6. Design and demonstrate simple quantum algorithmic solutions for problems in search, factorization, and optimization. (Bloom's Level: Create)

Course Relevance:

This course provides students with the foundational knowledge and practical skills required to understand quantum computation, develop quantum circuits, implement quantum algorithms using contemporary programming frameworks such as Qiskit or Q#, and evaluate their execution on

simulators and real quantum hardware. By integrating theoretical concepts with hands-on programming experience, the course prepares students to contribute to the rapidly evolving fields of quantum software engineering, cryptography, optimization, artificial intelligence, materials science, and scientific computing.

Syllabus
Theory

Unit 1: Quantum Complexity Theory (6 Hours)

Review of Quantum Computation Basics: Qubits, quantum gates, circuits, Tensor product, Dirac notation, measurement, Quantum entanglement and no-cloning theorem, Quantum circuit model and unitary evolution,

Classical complexity classes: P, NP, BPP, Quantum classes: BQP, QMA, Oracle problems and black-box lower bounds, Quantum query complexity

Unit 2: Deutsch and Deutsch–Jozsa Algorithm (6 Hours)

Problem setting and quantum solution, Oracle-based computation, Comparison with classical solutions, Circuit implementation,

Simon's Algorithm: Hidden subgroup problem, Exponential quantum advantage, Connection to Shor's algorithm, Fourier sampling over binary vectors

.

Unit 3: Grover's Search Algorithm (5 Hours)

Amplitude amplification and reflections, Grover iteration and search bounds, Applications to NP search problems, Variants of Grover's algorithm

.

Unit 4: Quantum Fourier Transform (5 Hours)

Quantum Fourier Transform (QFT): Definition and properties of QFT, Efficient circuit implementation of QFT, Role in quantum period-finding.

Shor's Factoring and Discrete Log Algorithm: Integer factorization using QFT, Order-finding problem, Discrete logarithms, Impact on RSA and cryptographic systems

Unit 5: Quantum Walk Algorithms (6 Hours)

Quantum Walk Algorithms : Quantum walks vs. classical random walks, Search algorithms on graphs, Element distinctness and hitting time.

Amplitude Estimation and Applications: Quantum phase estimation, Amplitude estimation algorithm, Applications in machine learning, finance, and simulation

Unit 6: Adiabatic and Variational Algorithms (5 Hours)

Quantum annealing and AQC, Variational Quantum Eigensolver (VQE), Quantum Approximate Optimization Algorithm (QAOA).

Syllabus
Laboratory

To gain hands-on experience with quantum circuits, quantum gates, and entanglement using IBM's Qiskit framework.

List of Tutorial

1. Simulate quantum algorithms using IBM Qiskit, Cirq, or Q#
2. Implement Deutsch-Jozsa, Grover's, and Shor's algorithm
3. Run circuits on real quantum devices via IBM Quantum Experience
4. Visualize QFT and phase estimation circuits
5. Use PennyLane or Qiskit for VQE and QAOA mini projects

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	0	0	1	0	0	0	0	0	0	2	3		2	
2	2	2	3	1	3	0	0	0	0	0	0	2	3		2	
3	2	2	3	2	3	0	0	0	0	0	1	2	3		2	
4	2	3	2	3	3	0	0	0	0	0	0	2	3		2	
5	2	3	2	3	2	0	0	0	0	0	1	3	3		2	
6	2	2	3	2	3	0	0	0	1	1	2	3	3		2	
Avg	3	2	0	0	1	0	0	0	0	0	0	2	**3**		**2**	

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Books and E-Resources

For Reference Print Book -

1. Michael Nielsen & Isaac Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010
2. Phillip Kaye et al., An Introduction to Quantum Computing, Oxford University Press, 2007
3. Ashley Montanaro, Quantum Algorithms – University of Bristol notes

For Reference Electronic Book –

1. Ronald de Wolf, Lecture Notes on Quantum Algorithms – Available online
2. Qiskit Textbook, Learn Quantum Computation Using Qiskit – https://qiskit.org/textbook

For MOOCs and other learning Resources

1. Quantum Computing with Qiskit and Advanced Algorithms-Coursera - https://www.coursera.org/learn/packt-quantum-computing-with-qiskit-and-advanced-algorithms-wils7

B. Tech. Final Year Computer Engineering AY 2026-27

Module VII and VIII
Course Content

Syllabus Template **FF No. : 654**

OE1: LinkedIn Learning

LinkedIn Learning Courses **Credits: 2**

Sr. No.	Course code	Specialization Name	Link
1	MD4274	Large Language Models Skill Development	https://www.linkedin.com/learning/paths/develop-your-skills-with-large-language-models?u=126888530
2	MD4275	Mastering Microsoft Power BI	https://www.linkedin.com/learning/paths/master-microsoft-power-bi-15399694?u=126888530
3	MD4276	Generative AI Skills for Developers	https://www.linkedin.com/learning/paths/building-generative-ai-skills-for-developers?u=126888530
4	MD4277	Career in Data Analysis	https://www.linkedin.com/learning/paths/explore-a-career-in-data-analysis?u=126888530
5	MD4278	Concepts of Data Visualization and Storytelling	https://www.linkedin.com/learning/paths/master-the-concepts-of-data-visualization-and-storytelling?u=126888530
6	MD4279	AWS Certified Solutions Architect	https://www.linkedin.com/learning/paths/prepare-for-the-aws-certified-solutions-architect-associate-saa-c03-certification?u=126888530
7	MD4280	IT Security Specialist	https://www.linkedin.com/learning/paths/become-an-it-security-specialist?u=126888530
8	MD4281	Technical Program Management	https://www.linkedin.com/learning/paths/technical-program-management?u=126888530
9	MD4282	Natural Language Processing Skill Development	https://www.linkedin.com/learning/paths/advance-your-skills-in-natural-language-processing?u=126888530
10	MD4283	Prompt Engineering Skills	https://www.linkedin.com/learning/paths/develop-your-prompt-engineering-skills?u=126888530
11	MD4284	Essentials in Generative AI	https://www.linkedin.com/learning/paths/career-essentials-in-generative-ai-by-microsoft-and-linkedin?u=126888530
12	MD4285	Python in Finance	https://www.linkedin.com/learning/paths/python-for-data-professionals-in-finance?u=126888530
13	MD4286	Understanding Quantum Computing	https://www.linkedin.com/learning/paths/understanding-quantum-computing?u=126888530
14	MD4287	Foundational Maths for Machine Learning	https://www.linkedin.com/learning/paths/foundational-math-for-machine-learning?u=126888530

Syllabus Template **FF No. : 654**

CS4217:: Human-Computer Interaction (HCI)

Credits: 02 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: NA

Course Objectives:

1. To categorize IT applications based on measurable human factors,
2. To study the user community through user survey and/or field visit,
3. To design user-friendly user interfaces with due consideration of interface theory and principles,
4. To apply usability evaluation methods to identify the usability issues with IT applications,
5. To understand the kind of documentation required for IT applications,
6. To integrate web and mobile app design approaches as per user requirements.

Course Relevance:

Human-Computer Interaction (HCI) is a multi-disciplinary socio-technical course, with a goal of bringing the power of computers and communication systems to users, customers or people. It aims to make all computing and communications systems more accessible, maintainable and useful in working, learning and recreational life of their users. It helps every computing, web or mobile application to become really user-centered, increasing its number of users as well as related sales.

Section 1: Topics / Contents

Unit-I Fundamentals of Human Computer Interaction: 05 Hours

Definition of HCI, Interdisciplinary Nature, Related Disciplines, Usability, Types of Usability, User Interface (UI), Measurable Human Factors, Accessibility, Differently-abled Users.

Unit-II Interaction Concepts and Models: 05 Hours

User Persona, User Categorization, Golden Rules of Interface Design, Miller's Principle, Task Analysis - GOMS, Contextual Inquiry, Work Models, Interaction Styles.

Unit-III Design Process: 04 Hours

Design Concept, Three Pillars of Design, Process of Design, Ethnographic Observations, Participatory Design, Internationalization.

Section 2: Topics/Contents

Unit-IV Usability Evaluation: 05 Hours

Expert-based Evaluation, User-based Evaluation, Formative Evaluation, Summative Evaluation, Heuristic Evaluation, Cognitive Walkthrough, Semiotic Analysis, Icon Categorization, User Surveys, Interviews, Usability Testing.

Unit-V Documentation and Groupware: 05 Hours

Classification of Documents, Reading from Displays, Online Help, Tutorials, Error / Warning Messages, Groupware, Computer Supported Cooperative Work (CSCW), Dimensions of Cooperation, Challenges with Online Communications.

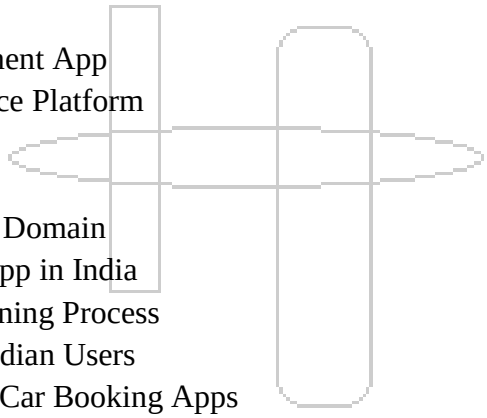
Unit-VI Website and Mobile App Design: 04 Hours

Content Design, Interaction and Navigation Design, Presentation Design, Differences in Design Approaches, Design and Evaluation Tools.

List of Home Assignments:

Design:

1. Social Network for Spiritual Users
2. App for Alzheimer's disease
3. Health Tracking App
4. Ration Card Management App
5. Innovative e-Commerce Platform



Case Study:

1. Chatbot in Healthcare Domain
2. Best Food Ordering App in India
3. Online Teaching-Learning Process
4. Use of Twitter with Indian Users
5. User Experience with Car Booking Apps

Blog

1. Noise of Notifications
2. Challenges in Food Delivery Service
3. Need for Accessibility Guidelines
4. Usability of Autonomous Vehicles
5. Failure of Usability Testing

Surveys

1. User Experience with Video-Conferencing Apps
2. User Errors on Social Networking Sites (SNS)
3. Challenges for Hearing Impaired Users with IT Applications
4. Most Popular Indian Mobile Apps (Made In/By India)

5. Impact of Ban on Chinese Apps in India

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books: ***(As per IEEE format)***

1. Ben Shneiderman, "Designing the User Interface", Third Edition, Pearson Education, ISBN 81-7808-262-4.
2. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human-Computer Interaction", Third Edition, Pearson Education, ISBN 81- 297-0409-9.

Reference Books: ***(As per IEEE format)***

1. Donald Norman, "The Design of Everyday Things", 2002 Edition, Basic Books, ISBN 100-465-06710-7.
2. Wilbert Galitz, "The Essential Guide to User Interface Design", Second Edition, Wiley-Dreamtech India (P) Ltd., ISBN 81-265-0280-0.
3. John Carroll, "Human-Computer Interaction in the New Millennium", Pearson Education, ISBN 81-7808- 549-6.

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com

Course Outcomes: Students will be able to

1. Appreciate the differences among IT applications and their categories based on measurable human factors.
2 Study the user community through user survey and/or field visit.
3. Design user-friendly user interfaces as per user requirements and UI design principles.
4. Apply a suitable usability evaluation method to identify the usability issues.
5. Understand the kind of documentation required for IT applications.
6. Enhance UI designs as per desired web or mobile app design approach.

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO 8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS4217.1	2	**3**	-	-	-	-	-	-	-	-	-	2	-	-	-	2
CS4217.2	2	3	-	**2**	-	-	-	-	-	-	-	2	-	-	-	2
CS4217.3	3	2	3	-	2	3	2	**2**	2	-	-	2	-	-	3	2
CS4217.4	2	3	3	2	-	-	-		-	-	-	2	-	-	2	2
CS4217.5	3	-	-	-	-	-	-	-	-	2	**1**	-	-	-	**2**	2
CS4217.6	3	2	3	2	2	3	2	2	2	-	-	-	-	-	**3**	**2**
Average	**2.5**	**2.6**	**3.0**	**2.0**	**2.0**	**3.0**	**2.0**	**2.0**	**2.0**	**2.0**	**1.0**	**2.0**	**-**	**-**	**2.5**	**2.0**

CO Attainment levels:

Weights for attainment levels: L1 - Easy- 0.75 L2 - Comfortable - 0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2 – L3, CO3 – L2, CO4 – L2, CO5 – L1 and CO6 – L3

Future Course Mapping:

User Interface Design, Usable Security & Intelligent User Interfaces

Job Mapping:

UI Designer, Product Designer, Software Engineer, Mobile App Developer and Other Jobs

Syllabus Template **FF No. : 654**

CS4209: Parallel Computing

Credits: 02 **Teaching Scheme Theory: 2 Hours/Week**

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 parallel computing
2. To understand various GPU Architecture.
3. To write CUDA programs for parallel implementation
4. To organize the memory management in GPU
5. 5.To optimize parallel programs on GPU using CUDA
6. To solve the scientific problems using GPUs

Course Relevance: Parallel 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 centers.

Section 1: Topics / Contents

Unit-I Introduction to Parallel Computing : 06 Hours

Motivating Parallelism, Scope of Parallel Computing, Parallelism vs Concurrency, Types and levels of parallelism, Flynn's classification, Amdhal's law; Parallel computer architectures : PRAM, Distributed memory systems,Shared memory systems and cache coherence. Concept of thread and process, programming parallel computers, Parallel computing architectures, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability

Unit-II Parallel Programming Model : 04 Hours

Common Unified Device Architecture (CUDA), CUDA programming model, Concept of grid, block and thread, thread index generation, warp, kernel & kernel launch

Unit-III CUDA APIs : 04 Hours

Programming for GPU's in C/C++ using CUDA APIs: Memory transfers, Writing and executing kernel functions, Writing device functions, Thread synchronization, Data Dependences and Race Conditions, Organizing Parallel Threads.

.

Section 2:Topics/Contents

Unit-IV GPU Architecture: 04 Hours

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 / Hopper

Unit-VMemory Organization and Optimization: 06 Hours

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.

Unit-VI Problem solving using GPUs : 04 Hours

Single vs. double precision, light weight scientific computing exercises, Image processing applications, Matrices etc .

List of Home Assignments:

Design:

6. Parallelizing Search Trees for Chess
7. Parallel Algorithm for Searching
8. Parallel Algorithm for sorting
9. Parallel Algorithm for Data mining
10. Parallel Algorithm for Image Processing

Case Study:

6. Nvidia DGX2
7. Jetson nano Developer Kit
8. GPU Accelerated Apache Spark
9. The Jetson Xavier NX Developer Kit
10. NVIDIA Ampere architecture

Blog

6. Cuda library
7. Turing mesh shaders
8. Low level GPU Virtual memory management
9. Memory Hierarchy of GPU
10. Comparison of Various GPUs

Surveys

6. Smart Hospitals through AI with GPUs
7. Clara Models to help fight with COVID 19
8. GPU Accelerated Molecular Dynamics Applications
9. Medical Imaging applications of GPU
10. Ray Tracing Applications of GPU

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

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-meiHwuCUDA: Programming Massively Parallel Processors: A Hands-On Approach. © ELSEVIER Inc. 3Jason 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, www.coursera.com, www.udemy.com, www.developer.nvidia.com

Course Outcomes: Students 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. Realize GPU Architecture.

5. Optimize the parallel programs on GPU using CUDA
6. Design and develop new solutions to research problems

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1			3													
CO2					3											
CO3							2									
CO4											1					
CO5												1				
CO6															3	
Average	**0**	**0**	**3.0**	**0**	**3.0**	**0**	**2.0**	**0**	**0**	**0**	**1.0**	**1.0**	**0**	**0**	**3.0**	**0**

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L3, CO2 – L3, CO3 – L2, CO4 – L1, CO5 – L1 and CO6 – L3

Future Course Mapping:
High Performance Computing , Distributed Computing

Job Mapping:
Full Stack Architect-GPU , HPC GPU Application Developer & Consultant , GPU Programming Professional , GPU Performance Analysis Lead / Architect ,
GPU Advocate Associate

Syllabus Template **FF No.: 654**

CS4222:Image Processing

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

Course Prerequisites: Digital logic Design, Microprocessor, Computer Organization.

Course Objectives:
1.To describe different color models and image processing techniques.
2.To analyze image condition and deduce enhancement algorithms.
3.To apply image segmentation to identify the region of interest
4.To develop an algorithm to recognize the specified objects in the given image.
5.To study different image morphological operation.
6.To learn different image compression techniques.

Course Relevance:
Vision sense is the most powerful human sense organ. In the world where intelligent automation is taking place, image processing is a vital domain for research and development. In Industry 4.0, image processing systems built around industrial cameras are an essential component in automated production. Throughout all steps of production, from the inspection of raw materials and production monitoring (i.e., flaw detection) to final inspections and quality assurance, they are an indispensable part of achieving high efficiency and quality standards. In the Entertainment Industry, latest trends such as 4K video streaming requires high quality compression that can provide limited/no loss image quality with high fps. In social networking, sharing images has been a vital part. Creating innovative effects and overall manipulating the images will be explored

Section 1: Topics/Contents

Unit-I Introduction **4 Hours**
Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YCbCr and some basic relationships between pixels, linear and nonlinear operations, Image sampling and quantization.

Unit-II Image Enhancements **5 Hours**
Memory-less operations, Spatial domain image enhancements: Denoising filters, Smoothing Operation, Sharpening Operation, and Contrast stretching /enhancement, histogram and histogram equalization.

Unit-III Image segmentation **5 Hours**

Classification of image segmentation techniques: Edge-based Segmentation, Region based techniques. Binarization: Global Thresholding, Adaptive thresholding. Types of Edge detector: derivative filters, Sobel, Canny. Edge linking. Feature Extraction.

Section2:Topics/Contents

Unit-IV Morphological Operation **4 Hours**

Binary Morphology, Erosion Dilation, Opening and Closing.

Unit-V Feature Extraction and Object Recognition **5 Hours**

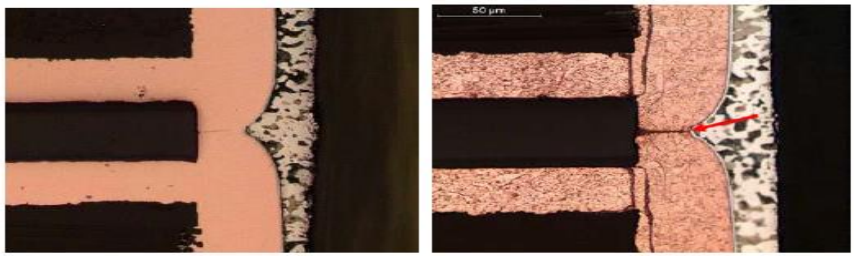
Feature points and feature detection (Line, circle and corner). Line detection: RANSAC, Hough Transform. Corner detection: Harris Corner Detector. Feature descriptors, Descriptor matching. SIFT, Boundary representation (Chain code), Boundary detection-based techniques.

Unit-VI Image Compression **5 Hours**

Introduction and need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, Run Length Coding, Huffman Coding, Shannon Fano coding).

List of Design based Home Assignments:

1.Design an algorithm to identify fault in a "PCB inspection system" as shown below



2.Design an algorithm to perform segmentation of the image below to extract the mango from its background.

3.Design an algorithm to get from image 1 to image 2.

4.Design an algorithm to recognize character “0” in the image below.

5.Design an algorithm to compress a 300x300 pixel image with horizontal black to white gradient as shown below.

List of Case Study based Home Assignments:

6. Cam-scanner: Document scanning app
7. Tesseract OCR library
8. Instagram filters
9. OpenCV
10. Google Street View

List of Blog based Home Assignments:

6. Image processing on Embedded platforms
7. Face recognition system security analysis for authentication
8. Image processing in MSME for effective automation
9. H.264 codec for image streaming
10. Role of mathematics in image processing

List of Survey based Home Assignments:

6. Image quality metrics
7. Vision based self-driving car safety
8. Compression techniques & codecs
9. State of the art applications such as AR/ XR
10. Human recognition in social networking apps like Facebook

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books: ***(As per IEEE format)***

1. *Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.*
2. *Anil K. Jain, "Fundamental of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.*

Reference Books: ***(As per IEEE format)***

1.*Pratt, "Digital Image Processing," Wiley Publication, 3rdEdition, ISBN 0-471- 37407-5.*
2. *K.R. Castleman, "Digital Image Processing,"3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.*

MOOCs Links and additional reading material:

1. https://nptel.ac.in/courses/117/105/117105135/
2. https://nptel.ac.in/courses/106/105/106105032/
3. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/
4. https://www.coursera.org/learn/computer-vision-basics

Course Outcomes:

The student will be able to –
1.Recognize different color models and image processing techniques. (1)
2.Select image enhancement algorithm to improve the quality of image. (2)

3.Build image segmentation techniques to identify region of interest. (4)
4.Predict image morphological techniques to resize the image. (3)
5.Construct an algorithm to recognize the specified objects in the given image. (5)
6. Identify different image compression techniques to reduce the size of image. (3)

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2		1	-	-	-	-	-	-	-	2	2	2	3	3
CO2	3	2	2	1	3	-	-	2	-	-	-	2	2	2	3	3
CO3	2	2	3	2	3	-	-	-	1	-	-	2	2	2	3	3
CO4	3	3	3	2	3	2	-	-	-	-	-	2	2	2	3	3
CO5	3	2		2	3	2	-	-	1	1	2	2	2	2	3	3
CO6	3	2		2	3	2	2	2		1	2	2	2	2	3	3
Average	**2.8**	**2.16**	**2.7**	**1.88**	**3**	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**2.0**	**2.0**	**2.0**	**2.0**	3.0	3.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L2, CO3 – L4, CO4 – L3, CO5 – L5 and CO6 – L3

Future Course Mapping:

1. Augmented Reality
2. Multimedia Processing

Job Mapping:

1. Augmented Reality Experience Designer
2. Automation Engineer
3. Embedded Software Developer
4. Image Processing Expert

Syllabus Template **FF No. : 654**

CS4275: Introduction to Machine Learning

(https://onlinecourses.nptel.ac.in/noc24_cs101/preview - offered by IIT Madras)

Credits: 2

Course Prerequisites:
Statistical Mathematics, Artificial Intelligence

Course Relevance:

With the increased availability of data from varied sources there has been increasing attention paid to the various data driven disciplines such as analytics and machine learning. In this course we intend to introduce some of the basic concepts of machine learning from a mathematically well motivated perspective. We will cover the different learning paradigms and some of the more popular algorithms and architectures used in each of these paradigms..

All students have to complete this course from NPTEL-SWAYAM under the guidelines provided by dean academics and to be coordinated by department coordinator assigned for this course.

Course Outcomes:

1. Understand the features of machine learning to apply on real world problems.-01
2. To develop various supervised machine learning algorithms in a wide range of real-world applications.-02
3. Understand a wide variety of Statistical learning algorithms and apply on different applications-03
4. Analyze the concept of Support Vector Machines, Naive Bayes Classifiers for learning linear and non-linear Classifiers.-05
5. To analyze Decision Tree algorithms to solve problems of real world -04
6. Demonstrate understanding of unsupervised learning, Reinforcement learning and their applications.-03

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	2	3	1	0	1	1	1	0	2	2	2	3	0
CO2	3	3	3	2	3	2	0	1	1	1	0	2	2	3	3	0
CO3	2	3	3	3	3	2	0	1	1	1	0	2	2	3	3	0
CO4	3	3	3	3	3	2	0	1	1	1	0	2	2	3	3	0
CO5	3	3	3	3	3	2	0	1	1	1	0	2	2	3	3	0
CO6	3	3	3	3	3	2	0	1	1	1	0	2	2	3	3	0
Average	**2.83**	**3**	**3**	**2.67**	**3**	**2**	**0**	**1**	**1**	**1**	**0**	**2**	**2**	**2.83**	**3.0**	**0**

Attainment Levels: 1,2,3,5,4,3

Syllabus Template **FF No. : 654**

CS4276: Deep Learning

(https://onlinecourses.nptel.ac.in/noc24_cs114/preview - offered by IIT Ropar**)**

Credits: 2

Course Prerequisites:

Statistical Mathematics, Artificial Intelligence, Machine Learning

Course Relevance:

The availability of huge volume of Image and Video data over the internet has made the problem of data analysis and interpretation a really challenging task. Deep Learning has proved itself to be a possible solution to such Computer Vision tasks. Not only in Computer Vision, Deep Learning techniques are also widely applied in Natural Language Processing tasks. In this course we will start with traditional Machine Learning approaches, e.g. Bayesian Classification, Multilayer Perceptron etc. and then move to modern Deep Learning architectures like Convolutional Neural Networks, Auto encoders etc. On completion of the course students will acquire the knowledge of applying Deep Learning techniques to solve various real life problems.

All students have to complete this course from NPTEL-SWAYAM under the guidelines provided by dean academics and to be coordinated by department coordinator assigned for this course.

Course Outcome:

1. Demonstrate understanding of a 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. Implement convolutional networks for image recognition/classification tasks.
6. Demonstrate Understanding of Recurrent nets and their applications.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	2	3	1	0	1	1	1	0	2	0	3	3	0
CO2	3	3	3	2	3	2	0	1	1	1	0	2	0	3	3	0
CO3	2	3	3	3	3	2	0	1	1	1	0	2	0	3	3	0
CO4	3	3	3	3	3	2	0	1	1	1	0	2	0	3	3	0
CO5	3	3	3	3	3	2	0	1	1	1	0	2	0	3	3	0
CO6	3	3	3	3	3	2	0	1	1	1	0	2	0	3	3	0
Average	3	3	3	2.67	3	1.83	0	1	1	1	0	2	0	3	3	0

Attainment Levels: 3,3,5,4,5,4

Syllabus Template **FF No. : 654**

CS4225, CS4226: Major Project

Credits:.13..... **Teaching Scheme Theory: …26... Hours/Week**

Course Prerequisites: Project Based Learning

Aim

This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

- Transform existing Ideas into conceptual models.
- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- Apply the results from physical models to create real target systems.

Project Group and Topic Selection and Synopsis:

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 student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. 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.

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 based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The 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.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.

6. Upon project completion, the Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents : (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

Sr.	Project Item
1	Project Cover Front Page
2	Project Completion Certificate [Institute]
3	Project Completion Letter [In case of Sponsored Projects]
4	Acknowledgments
5	Table of Contents
6	List of Figures
7	List of Tables
8	Project Synopsis [Problem Background, Existing System Details, Proposed Solution]
9	Feasibility Study Report
10	Project Plan
11	System Requirement Specification
12	System Analysis Document: UML Use Case Diagrams
13	System Analysis Document: UML Sequence Diagrams
14	System Analysis Document: UML State Diagrams
15	System Design Document with Module Specifications

16	System Implementation
17	System Testing and Experimental Findings
18	Conclusion
19	References

7. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration 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 and overall development effort taken by the candidates.

Note:
The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment: MSE and ESE
1. Mid Semester Assessment – 50 Marks to be converted to 30 Marks.
2. End Semester Assessment – 100 Marks to be converted to 70 Marks.

Mid Semester Assessment

Sr. No.	Parameter	Marks
1	Problem Statement	10
2	Literature Review	10
3	Group formation and identification of individual responsibility	10
4	Objective of Project activity	10
5	Knowledge of domain, latest technology and modern tools used /to be used	10
	TOTAL	**50**

End Semester Assessment

Sr. No.	Parameter	Marks
1	Realization of project as per problem statement	10
2	Design, Testing / Experimentation, Analysis / Validation	30
3	Documentation and Report Writing	20
4	Quality of Work	15
5	Performance in Question & Answers Session	15
6	Regular interaction with guide	10
	TOTAL	**100**

Course Outcomes:

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

CO1: Identify the real life problem from societal need point of view

CO2: Prepare the requirement engineering, feasibility analysis documents

CO3: Form the teams and share responsibilities according to individual skill strengths

CO4: Create design documents to build software solutions

CO5: Develop software solutions based on standard engineering specifications

CO6: Perform the verification and validation up to the mark

CO PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3		2					3			3				3
CO2	2	3	3	2	2				3	3	2	3	3		3	3
CO3	2	-	-	-	-				3		2	3	3			3
CO4	2	3	3	2	2	3	3.0	2.0	3	3	2	3	3	3	3	3
CO5	2	3	3	2	2				3	3	2	3	3	3	3	3
CO6	2	2	2	3	2				3	2	2	3	3	2	3	3
Average	**2.0**	**2.8**	**2.75**	**2.83**	**2.0**	**3.0**	**3.0**	**2.0**	**3.0**	**2.75**	**2.0**	**3.0**	**3.0**	**1.75**	**3.0**	**3.0**

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Syllabus Template **FF No. : 654**

CS4292: Design Thinking 7

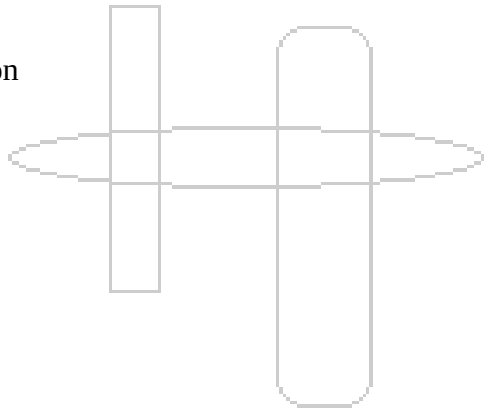
Credits: 01 **Teaching Scheme: Tutorial 01 Hr/week**

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:
To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship



Section 2: Topics/Contents

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

Course Outcome: **[**Publication of paper or patent]
The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	3
CO2	1	1	1	1	1	--	--	--	--	--	--	1	2	1	1	3
CO3	2	2	3	3	2	2	1	2	2	3	--	1	2	2	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	-	-	2	3
CO5	1	1	1	1	1	--	--	--	--	--	--	1	-	-	1	2
CO6	2	2	2	2	2	2	1	3	2	3	--	1	2	2	2	3
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	1
Average	**1.57**	**1.57**	**1.71**	**1.71**	**1.57**	**2.0**	**1.0**	**2.33**	**2.0**	**3.0**	**1.0**	**1.0**	1.66	1.66	1.71	2.5

Syllabus Template **FF No. : 654**

CS4236, CS4238, CS4232, CS4202: Industry Internship, Research Internship

Credits:.15..... **Teaching Scheme Laboratory: …32... Hours/Week**

Guidelines:

Internships are educational and career development opportunities, providing practical experience in a field or discipline. Internships are far more important as the employers are looking for employees who are properly skilled and having awareness about industry environment, practices and culture. Internship is structured, short-term, supervised training often focused around particular tasks or projects with defined time scales. Core objective is to expose technical students to the industrial environment, which cannot be simulated/experienced in the classroom and hence creating competent professionals in the industry and to understand the social, economic and administrative considerations that influence the working environment of industrial organizations. Engineering internships are intended to provide students with an opportunity to apply theoretical knowledge from academics to the realities of the field work/training.

The following guidelines are proposed to give academic credits for the internship undergone as a part of the B.Tech. Engineering curriculum.

Duration:

Industry Internship will be started at the beginning of the semester 7 or semester 8 or yearlong for the duration 6 months or 12 months.

Identification of Internship work:

Student may choose to undergo Internship at Industry/Govt./NGO/MSME/ Innovation/IPR/Entrepreneurship. Contacting various companies for Internship and Internship work identification process should be initiated at the end of 6th semester in coordination with training and placement cell/ industry institute cell/ internship cell. This will help students to start their internship work on time. Students can take internship work in the form of online/onsite work from any of the following but not limited to:

- Working for consultancy/ research project
- Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council/ start-ups cells of institute
- Industry / Government Organization Internship,
- Internship through Internshala

- In-house product development, intercollegiate, inter department research internship under research lab/group, micro/small/medium enterprise/online internship
- Research internship under professors, IISC, IIT's, Research organizations

Internship Documents Submission:

Students must submit internship offer letter, internship completion letter, FF 1029 (Students feedback form), FF 1030 (Industry feedback about interns).

Students must present their internship progress time to time to faculty mentors. Faculty mentors and industry mentors both can evaluate the progress of the intern combiningly.

Internship Work Evaluation:

In-semester and end semester internship evaluation and assessment will be done by internal (Faculty mentor) and external examiners - a supervisor from industry.

After completion of Internship, the student should prepare a comprehensive report to indicate what he has observed and learnt in the internship/training period. The student may contact Industrial Supervisor/ Faculty Mentor/TPO for assigning special topics and problems and should prepare the final report.

If the student remain absent without prior intimation to the department/institute/concern authority/T & P Cell, his entire training can be cancelled and he will fail.

Course Outcomes: Industry Internship

On the completion of course, students will able to-

1. Understand real-world applications, workplace environment and operating procedures
2. Adapt skill for learning and applying modern tools and technologies
3. Apply professional values and ethical standards
4. Perform as an individual and as a team member effectively to changing conditions
5. Encompass improved writing, verbal communication and documentation skills
6. Learn about career positions and occupations along with the qualities and training required to obtain those positions

CO-PO Map: Industry Internship

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3		2					3			3				2
CO2	2	3	3	2	3	2	3	2	3		2	3	3	3	3	3
CO3	2	3	3	2				2	3		2	3	3	3	3	3
CO4	2								3			3				
CO5	2		2		3			2	3	3	2	3			3	
CO6	2					2			3			3				
Average	2.0	3.0	2.66	2.0	3.0	2.0	3.0	2.0	3.0	3.0	2.0	3.0	3.0	3.0	3.0	2.66

CO attainment levels:

CO1 – 3, CO2 –5, CO3 –2, CO4 –2, CO5 – 3, CO6 –2

Course Outcomes: Research Internship

On the completion of course, students will able to-

1. Develop an ecosystem to promote entrepreneurship and research culture among the students.
2. Learn first-hand to apply techniques, resources, modern engineering tools for prediction modelling to complex engineering activities.
3. Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and consequent responsibilities.
4. Perform as an individual and as a team member.
5. Understand Engineering and Management Principles.
6. Exercise R & D aptitude focusing on the knowledge creation and dissemination through engineering artifacts creation, construction and presentation.

CO-PO Map: Research Internship

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2											3				3
CO2	3	3	3	2	3	2	2	2				3	3	3	3	3
CO3	3	3	3	2		3						3	3		3	3
CO4									3			3				3
CO5	3	2	3	2								3	3		3	3
CO6	3	3	2	3			3	2.0		3.0	2.0	3			3	3
Average	2.8	2.75	2.75	233	3.0	2.5	2.5	2.0	3.0	2.75	2.0	3.0	3.0	3.0	3.0	3.0

CO1 – 3, CO2 –4, CO3 –5, CO4 –2, CO5 – 3, CO6 –4