

www.sjcit.ac.in

||Jai Sri Gurudev||

Sri Adichunchanagiri Shikshana Trust (R.)

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

AICTE Approved, Accredited by NAAC with A+ Grade & NBA (CSE, ISE, ECE, ME & CV), NIRF(151-300) in 2023,Gold Rated by QS I-Gauge

P.B. No.20, B.B Road, Chikkaballapur - 562 101, Karnataka

Estd : 1986

I & II Semester
Scheme & Syllabus of M.Tech Programme
Computer Science & Engineering
(2026 Scheme)

Core Values

Spiritual, Judicious, Commitment, Integrity, Trustworthy



AICTE | NIRF NATIONAL INSTITUTIONAL RANKING FRAMEWORK 151-300 | NBA | A+ NAAC | QS I-GAUGE GOLD | INSTITUTION'S INNOVATION COUNCIL (Ministry of HRD Initiative) 4-Star Rating



www.sjcit.ac.in

||Jai Sri Gurudev||

Sri Adichunchanagiri Shikshana Trust (R.)

SJC INSTITUTE OF TECHNOLOGY

(An Autonomus Institute Under VTU Belagavi & Approved by AICTE, New Delhi)

Accredited by NAAC with A+ Grade & NBA (CSE, ISE, ECE, ME & CV), NIRF (151-300), QS-I Gauge GOLD Rated

Post Box #20, B.B. Road, Chikkaballapur - 562 101, Karnataka

Estd.:1986

Scheme of Teaching and Examinations (2026)

M.Tech.(Computer Science and Engineering)

Choice-Based Credit System (CBCS) and Outcome-Based Education (OBE)

I SEMESTER (Common to all specialized Programmes under one stream)

I SEMESTER

Sl. No	Courseand Course Code		Course Title	Teaching Department (TD) and Question Paper Setting Board(PSB)	Teaching and Learning Scheme					Examination				Credits
					Classroom Instruction (Cl)hoursper semester		Laboratory Instruction (LI) hours Per semester	Term work s self -learning in hours	Teaching -Learning hours per semester = Cl +LI+TW+SL	Durationin h ours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	TW+SL						
1	MAT/PCC	1MCS101	Advanced Mathematics	Maths	42	0	0	48	90	3	50	50	100	3
2	PCC	1MCS102	Advanced Data Structures and application	CSE	42	0	0	48	90	3	50	50	100	3
3	PCC	1MCS103	Data Science and Management	CSE	42	0	0	48	90	3	50	50	100	3
4	IPCC	1MCS104x	Integrated Professional Core Course-I	CSE	42	0	28	50	120	3	50	50	100	4
5	PEC	1MCS105x	Professional Elective Course-I	CSE	42	0	0	48	90	3	50	50	100	3
6	PEC	1MCS106x	Professional Elective Course-II	CSE	42	0	0	48	90	2	50	50	100	3
7	PCCL	1MCSL107x	Professional Core Courses Labs	CSE	--	--	28	2	30	03	50	50	100	1
8	NCMC	1MRMI108	Research Methodology and IPR		Online course (online.vtu.ac.in)									PP
								Total	**600**	--	**350**	**350**	**700**	**20**

Professional Elective Courses (PECs): Professional Elective Courses – PEC–I and PEC–II – are common to all branches of specialization within a particular Engineering stream. Students may choose the most appropriate elective based on their field of specialization and academic requirements. Note: The number of courses listed under each PEC group may exceed four, depending on the specializations under one stream.

Integrated Professional Core Courses (IPCC): The 1Mxx104x Group comprises specialization-specific core courses that are integrated with a practical component, ensuring application-oriented learning aligned with industry and research needs. The number of courses in the group depends on the number of specializations offered under a particular engineering stream.

Integrated Professional Core Course (IPCC)	
Code	**Title of the Course**
1MCS104A	Advanced Database Management Systems
1MCS104B	Advances in Computer Networks
1MCS104C	Advances Operating Systems
1MCS104D	Advanced Bigdata Analytics

Professional Elective Courses-I	
Code	**Title of the Course**
1MCS105A	Artificial Intelligence for IoT
1MCS105B	Artificial Intelligence for Cyber Security
1MCS105C	Advanced Computer Vision
1MCS105D	Data Analytics for Fraud Detection

Professional Elective Courses-II	
Code	**Title of the Course**
1MCS106A	Software Engineering and Web Applications
1MCS106B	Advances in Block chain Technology
1MCS106C	Mobile and Web Application Security
1MCS106D	Data Engineering

Professional Core Courses Lab (PCCL)	
Code	**Title of the Course**
1MCSL107A	Software Automation Testing Laboratory
1MCSL107B	Computer Vision Laboratory
1MCSL107C	Advanced Data Structures Laboratory
1MCSL107D	Microservices Development Laboratory

Specialization in -(Computer Science and Engineering)

II SEMESTER

Sl. No	Courseand Course Code		Course Title	Teaching Department (TD) and Question Paper Setting Board(PSB)	Teaching and Learning Scheme					Examination				Credits
					Classroom Instruction (CI)hoursper semester		Laboratory Instruction (LI) hours Per semester	Term work s self -learning in hours	Teaching -Learning hours per semester = CI +LI+TW+SL	Durationin h ours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	TW+SL						
1	PCC	1MSCS201	Deep Learning and its Applications	CSE	42		0	48	90	03	50	50	100	3
2	PCC	1MSCS202	Cryptography and Network Security	CSE	42	0	0	48	90	03	50	50	100	3
3	PCC	1MSCS203	High Performance Computing	CSE	42	0	0	48	90	03	50	50	100	3
4	IPCC	1MSCS204	Advanced Algorithms	CSE	42	0	28	50	120	03	50	50	100	4
5	PEC	1MSCS205x	Professional Elective Courses-III	CSE	42	0	0	48	90	03	50	50	100	3
6	PEC	1MSCS206x	Professional Elective Course-IV	CSE	42	0	0	48	90	03	50	50	100	3
7	PCCL	1MSCSL207	Professional Core Course- Lab (AEC Lab)	CSE	--	--	28	2	30	03	50	50	100	1
8	PCC	1MSCSS208	Minor Project /Ability Skill Development Project*	CSE	--				60	03	50	50	100	2
								Total	**660**	--	**400**	**400**	**800**	**22**

***** An Ability Enhancement Course Laboratory is a practical, skill-oriented lab course designed to strengthen students' practical abilities, professional competencies that support communication, environmental awareness, computational thinking, interdisciplinary learning, and application skills through hands-on learning experiences. The laboratory may pertain to disciplinary or interdisciplinary involving experiments, design tasks, and mini-projects aligned with current industry practices.

Professional Elective Courses (PECs)				
PEC-III			PEC-IV	
Code	Title of the Course		Code	Title of the Course
1MSCS205A	Quantum Computing		1MSCS206A	Software Defined Networks
1MSCS205B	Fog and Edge Computing		1MSCS206B	Advanced Generative Models
1MSCS205C	Genetic Algorithms		1MSCS206C	Prompt Engineering
1MSCS205D	Secure Coding		1MSCS206D	Artificial Intelligence for Digital Twinning

Overview of Courses, Credits, Projects, and Internships under M.Tech. - 2026 Curriculum

(A) Abbreviations Adopted in the Scheme of Teaching and Examinations

Course		General Terms		General Terms	
Acronyms	**Expanded Form**	**Acronyms**	**Expanded Form**	**Acronyms**	**Expanded Form**
AEC	Ability Enhancement Course	AICTE	All India Council of Technical Education	SEE	Semester End Evaluation
ASC	Applied Science Course	CCA	Continuous Comprehensive Assessment	SGPA	Semester Grade Point Average
BSC	Basic Science Course	CGPA	Cumulative Grade Point Average	SL	Self-Learning
HSMC	Humanities Studies and Management Course	CI	Classroom Instruction	SWAYAM	Study Webs of Active-Learning for Young Aspiring Minds
IPCC	Integrated Professional Core Course	CIE	Continuous Internal Evaluation	T	Tutorial
NCMC	Non-Credit Mandatory Course	COE	Centre for Online Education	TW	Term Work
OEC	Open Elective	CUL	Cultural	VTU online	VTU online courses
PCC	Professional Core Course	NCrF	National Credit Framework		
PCCL	Professional Core Course	L	Lecture		
PEC	Professional Elective	LI	Laboratory Instruction		
PE	Physical Education	NPTEL	National Programme for		
SEC	Skill Enhancement	NSS	National Service Scheme		
YOG	Yoga	P	Practical		

(B) Academic Credit Framework in accordance with the National Credit Framework (NCrF)

The National Credit Framework (NCrF), introduced under India's education reforms, serves as an umbrella framework for the creditisation of learning across school education, higher education, vocational education, and experiential learning.

NCrF credit levels are determined based on a combination of cumulative years of learning, the complexity of learning outcomes, and the rigor of assessment. These levels signify the degree of competence and complexity attained by the learner, progressing from foundational schooling to advanced doctoral research.

Levels 1 to 4 correspond to school education up to the 12th standard, while Levels 4.5 to 8 correspond to higher education, ranging from the undergraduate level to the Ph.D. level.

In the earlier system, credits were primarily based on contact time (lecture hours per week × number of weeks per semester/year), assuming that learning is directly proportional to time spent in classroom instruction.

However, the NCrF adopts a comprehensive and outcome-based approach by assigning credits to all forms of learning, including, (i) Classroom teaching, (ii)Laboratory work, (iii)Projects, (iv)Internships and apprenticeships, (v)Online and blended learning, (vi)Fieldwork and community engagem ent, (vii)Recognition of Prior Learning (RPL), (viii)Sports and arts activities, and (ix)On-the-job training.

Credits are thus based on notional learning hours, representing the total learner effort rather than only contact hours.

This definition aligns Indian education with Global outcome-based credit systems, and international credit transfer practices [e.g., European Credit Transfer and Accumulation System (ECTS)-like frameworks].

According to NCrF, one academic year of learning is equal 1,200 notional learning hours (not a scientifically-validated law or psychological constant, but a policy design hours) and includes classroom teaching, self-study, assignments, projects, laboratory work, co-curricular and extra-curricular activities, internships, apprenticeships and other assessed learning experiences. For these 1,200 hours a student is expected to earn 40 credits, which means 1 credit corresponds to approximately 30 hours of academic learning.

Although traditional university systems often equated 1 credit with approximately 13 to15 contact hours per semester, VTU, in alignment with NCrF, considers 14-weeks of structured academic interaction per semester with students is equivalent to approximately 30 notional hours per credit.

The learning methodologies include:

(i) Classroom Instruction (CI): Includes different instructional/implementation strategies, such as Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc., to deliver theoretical concepts within the classroom measured in Number of hours per semester.

(ii) Laboratory Instruction (LI): Expressed as number of hours per semester which Includes experiments/practical performances / problem-based experiences in laboratory, workshop, field or other locations using different instructional / Implementation strategies.

(iii) Term work (TW): Includes assignments, seminars, presentations, case studies, micro projects, field activities, industrial visits, academic preparation duration and any other student activities in Number of hours per semester.

(iv) Self-Learning (SL): MOOCs (SWAYAM/NPTEL/Industry certified courses), spoken tutorials, online educational resources, self-initiated projects, Learning through digital resources etc in Number of hours per semester.

VTU, in alignment with NCrF, considers 14-weeks of structured academic interaction per semester with students as equivalent to approximately 30 notional hours per credit.

Table below shows the credit structure, equivalent NCrF learning hours adopted by VTU:

Table 7.0.1 VTU adopted NCrF credit structure

Particulars	Classroom Instruction (CI) in Hours per semester		Laboratory Instruction (LI) in Hours per semester	Term work (TW) and Self-Learning (SL) in Hours per semester	Total number of Teaching-Learning hours per semester
	L	T	P	TW + SL	
1 Credit theory course	14	-	-	16	30
1 Credit Practical course	-	-	28	02	30
2 Credit theory course	28	-	-	32	60
3 Credit theory course	42	-	-	48	90
3 Credit theory course with Tutorial	28	28	-	34	90
4 Credit theory course	56	-	-	64	120
4 Credit theory course with Tutorial	42	28	-	50	120
4 Credit Integrated Course	42	-	28	50	120
Project work	-	-	-	As per the type of project	
Internship	-	-	-	As per the type of Internship	

NCrF is designed to work with ABC for years, where, credits are digitally stored to accumulate, transfer, and redemption across institutions for vertical and horizontal mobility across sectors, which is essential for MEME.

(C) Details of Courses

(1) Integrated Professional Core Course (IPCC): The Integrated Professional Core Course (IPCC) refers to a core theory course that is integrated with a laboratory of the same subject. Each IPCC carries 4 credits, with Teaching–Learning hours structured (L : T : P) as either (3:0:2). The theory component of the IPCC shall be evaluated through both Continuous Internal Evaluation (CIE) and Semester End Examination (SEE). The laboratory part shall be assessed exclusively through CIE, with no SEE. However, questions derived from the laboratory part may be included in the SEE question paper to ensure comprehensive evaluation.

(2) Non-Credit Mandatory Courses (NCMC): are aimed at enhancing students' knowledge, skills, and awareness beyond the core curriculum. Successful completion of the NCMC is compulsory for fulfilling the requirements of the academic program. It shall not be considered for the computation of SGPA, CGPA

and vertical progression. Each student shall register for the prescribed NCMC(s) in the prescribed semester. A student who fails to qualify in the prescribed NCMC shall not be eligible for the conferment of the degree.

(3) Professional Elective Courses (PEC): A professional elective course (PEC) is intended to enhance the depth and breadth of educational experience in the Engineering and Technology curriculum of the same discipline.

(4) Open Elective Courses (OEC): A open elective course (OEC) is a course offered by departments other than a student's parent department. These interdepartmental /interdisciplinary courses allow students to explore disciplines beyond their core area of study. These courses are intended to promote interdisciplinary learning, broad-based education, thereby enhancing a student's overall knowledge, creativity, and employability. Registration to open electives shall be documented under the guidance of the Program Coordinator/ Advisor/Mentor/Proctor.

(5) Ability/Skill Enhancement Course Laboratory (AEC): An Ability Enhancement Course Laboratory is a practical, skill-oriented lab course designed to strengthen students' practical abilities, professional competencies that support communication, environmental awareness, computational thinking, interdisciplinary learning, and application skills through hands-on learning experiences.

The laboratory may pertain to disciplinary or interdisciplinary involving experiments, design tasks, and mini-projects aligned with current industry practices.

(6) Skill Enhancement Courses (SEC): These courses are intended to develop specific practical skills and competencies that improve students' employability, technical proficiency, and professional readiness to bridge the gap between academic and industry requirements. These courses emphasize hands-on training, application of theoretical knowledge, and development of discipline-relevant and transferable skills required in industry and society, and develop entrepreneurship and start-up skills.

(7) Online Courses: Online courses are educational programs delivered over the Internet through a digital platform, allowing students to access lessons, assignments, and discussions from anywhere at any time. Most online courses offer flexibility, allowing students to access materials and complete assignments on their own schedule. However, students have to pass the course within a stipulated period as per the norms of the university.

(8) VTU Online Courses: VTU Online Courses are online courses offered by the Centre of Online Education (COE), Mysuru. A wide range of multidisciplinary courses is made available to learners anytime and anywhere, enabling them to earn University-prescribed credits through duly conducted proctored examinations for the award of the degree.

NPTEL/SWAYAM courses adopted by VTU shall be offered in accordance with the NPTEL/SWAYAM course delivery framework, wherein teaching–learning is carried out through the online MOOC platform, assignments are administered periodically, and assessment is completed through a centrally conducted proctored examination as prescribed by the respective platform.

(9) NPTEL/SWAYAM Online Courses: The National Programme on Technology Enhanced Learning (NPTEL)/SWAYAM (Study Webs of Active Learning for Young Aspiring Minds) are the specific Indian platforms to host national Massive Open Online Courses (MOOCs). It offers online courses on a wide range of disciplines to learners anywhere, anytime, to earn university-prescribed credits through proctored examination for the award of a degree. All NPTEL/SWAYAM courses are MOOCs, but not all MOOCs are offered on these specific Indian platforms.



Semester: I					
Advanced Mathematics					
Course Code	**:**	**1MCS101**	**Credits**	**:**	3
Hours/Week (L: T: P:SL)	**:**	42: 0: 0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Discrete Mathematics Structure
Knowledge on Fundamental Concepts	Probability, Matrices
Knowledge on Connected Tools (If any)	Basic computer knowledge

Course Learning Objectives: This course will enable the students to	
CLO1	Advance learning in mathematical logic frameworks and propositional structures to construct and verify formal proofs.
CLO2	Develop a strong learning foundation in algebraic set operations to model and solve real-world problems.
CLO3	Enhance student learning of equivalence relations, partial orders, and invertible functions for logical and computational problem-solving.
CLO4	Learn graph theory properties, types, and isomorphic characteristics to analyze structural variations in networks.
CLO5	Describe the concept of random variables, probability distributions, specific discrete and continuous distributions with practical application in Computer Science Engineering and real-life situations.

Module – 1	**10 Hrs**
Mathematical Logic and Proof Techniques Mathematical Logic, Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates, Introduction to Proofs.	
Module – 2	**10 Hrs**
Relations and Functions Basic Structures: Sets, Cartesian product of sets, relations, Zero-one matrices and directed graphs, properties of relations, Equivalence relations. Partial orders, Function- types and properties, Composition of function, Invertible Function	
Module – 3	**10 Hrs**
Combinatorics and Matrices Operations on sets, Laws of set theory, countable and uncountable sets, addition principle, Principle of Inclusion and Exclusion, Pigeonhole principle. Solution for the system of equation by Cramer`s rule. Matrices: Eigenvalues and Eigenvectors.	
Module – 4	**10 Hrs**
Graph Theory Introduction to Graphs: Types of graphs, Incidence and degree, Paths and circuits, Isomorphism, sub-graphs, walks, paths and circuits, Shortest-Path Problems, connected graphs, disconnected graphs and components. Eulerian and Hamiltonian graphs: Euler graphs, Hamiltonian paths and circuits, Travelling salesman problem. Directed graphs, types of digraphs, Digraphs and binary relation, Graph Coloring: Coloring- Chromatic number.	

Module – 5	**10 Hrs**
Probability Distribution **Random variable and probability distribution** Concept of random variable, discrete probability distributions, continuous probability distributions, Mean, variance and co-variance, Binomial and Poison distribution.	

Topics for Self learning		
Module-1	**Applications:**	Boolean search engines, if-else conditions in coding, SQL WHERE clauses, everyday decision making
	Self-Study:	Logical puzzles, argument validation
Module-2	**Applications:**	Database table joins (union/intersection), Social media friend lists (relations)
	Self-Study:	Relation Matrices and Graph Representation of Relations, Wars hall's Algorithm for Transitive Closure
Module-3	**Applications:**	Password validation (bijective functions), Fibonacci in nature (recurrence), Error detection in QR codes & barcodes (Hamming/parity check)
	Self-Study:	Hamming distance concept.
Module-4	**Applications:**	Google Maps shortest path (connectivity), Social networks friend Connections (graphs).
	Self-Study:	Graph Coloring, Network Flow and Connectivity Analysis
Module-5	**Applications:**	Optimizing mail/garbage collection delivery routes, Sports league Scheduling.
	Self-Study:	Fleury's Algorithm (Eulerian circuits), Vizing's Theorem.

Course Outcomes: After completing the course, the students will be able to	
CO1	Analyze propositions and mathematical logic structures to construct and verify formal proofs.
CO2	Apply algebraic operations on sets to solve real-world problems.
CO3	Apply the properties of equivalence relations, partial orders, and invertible functions to solve computational and logical problems.
CO4	Analyze the structural properties, types, and isomorphic characteristics of various graph networks.
CO5	Evaluate probability distributions and random variable concepts to model and Simulate uncertainties in engineering systems.

Text Books	
1	Discrete Mathematics and Its Applications – Kenneth H. Rosen, 9th Edition, McGraw-Hill Higher Education, 2025. ISBN: 978-1-266-19154-1.
2	Graph Theory with Applications to Engineering and Computer Science – Narsingh Deo, Dover Edition, Dover Publications, 2016. ISBN: 978-0-486-80793-5.
3	Graph Theory – J.A. Bondy and U.S.R. Murty, 1st Edition, Springer London, 2008. ISBN: 978-1-846-28969-9.

Reference Books

1	Graph Theory – Frank Harary, Reprint Edition, Narosa Publishing House, 2022. ISBN: 978-81-85015-55-2
2	J.K. Sharma, 4th Edition, Laxmi Publications, 2022. ISBN: 978-93-51381-43-3.
3	A First Course in Graph Theory *(Successor to "Introduction to Graph Theory")* – Gary Chartrand and Ping Zhang, Revised & Corrected Edition, Dover Publications, 2012. ISBN: 978-0-486-48368-9..

Web Links and Video Lectures (e-Resources)	
1	NPTEL – Discrete Mathematical Structures **(**IIT Madras**),** YouTube Playlist**:** https://www.youtube.com/playlist?list=PL0862D1A947252D20
2	Coursera **–** Discrete Mathematics **(**University of California San Diego**),** Online Course**:** https://www.coursera.org/specializations/discrete-mathematics
3	MIT Open Courseware – Mathematics for Computer Science (6.1200J), Spring 2024: https://ocw.mit.edu/courses/6-1200j-mathematics-for-computer-science-spring-2024/

Semester: I					
Advanced Data Structures and Applications					
Course Code	**:**	1MCS102	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Basic data structures (arrays, linked lists, stacks, queues, trees, and graphs)
Knowledge on Fundamental Concepts	Elementary algorithms and complexity analysis
Knowledge on Connected Tools (If any)	Basic programming in C/C++/Java/Python

Course Learning Objectives: This course will enable the students to	
CLO1	Explain heap and priority queue structures, and hashing techniques.
CLO2	Construct balanced and multiway search trees for efficient data storage.
CLO3	Apply tries and augmented structures for fast data retrieval.
CLO4	Apply advanced data structures to solve real-world data problems.

Module – 1	**08 Hrs**
Fundamental Heap Structures and Hashing: **Heap Structures:** Heaps, Maintaining the Heap Property, Building a Heap, Priority Queues. **Hash ng Fundamentals:** Direct Address Tables, Hash Tables, Hash Functions	
Module – 2	**08 Hrs**
Advanced Hashing and Priority Structures: **Advanced Hashing:** Open Addressing, Perfect Hashing. **Dynamic Hashing:** Dynamic Hashing, Bloom Filters. **Advanced Priority Structures:** Pairing Heap, Symmetric Min-Max Heaps, Interval Heaps.	
Module – 3	**08 Hrs**
Balanced Trees and Mergeable Heap Structures: **Red-Black Trees:** Properties of Red-Black Trees, Rotations, Insertion, Deletion. **Mergeable Heap Structures:** Leftist Trees, Binomial Heaps, Fibonacci Heaps.	
Module – 4	**08 Hrs**
Search Trees and Trie Structures: **Adaptive and Multiway Search Trees:** Splay Trees, m-Way Search Trees, B-Trees, B+ Trees. **Trie Structures:** Digital Search Trees, Binary Tries and Patricia Trees, Multiway Tries.	
Module – 5	**08 Hrs**

Augmented and Specialized Structures:

Augmented Data Structures: Dynamic Order Statistics, How to Augment a Data Structure, Interval Trees.

Applied Trie Structues: Suffix Trees, Tries in Internet Packet Forwarding.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain heap and priority queue structures, and hashing techniques.
CO2	Construct balanced and multiway search trees for efficient data storage.
CO3	Apply tries and augmented structures for fast data retrieval.
CO4	Apply advanced data structures to solve real-world data problems.

Textbooks	
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 4th Edition, (2022), MIT Press.
2	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition (2023), Universities Press(India) Private limited.

Reference Books	
1	Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition, (2014) Pearson Education.
2	Sartaj Sahni, Data Structures, Algorithms and Applications in C++, 2nd Edition, (2011), Universities Press.
3	Peter Brass, Advanced Data Structures, 1st Edition, (2011), Cambridge University Press.

Web links and Video Lectures (e – Resources)	
1	NPTEL – Advanced Data Structures: https://nptel.ac.in/courses/106102064
2	MIT OpenCourseWare – Advanced Data Structures: https://ocw.mit.edu/courses/6-851-advanced-data-structures-spring-2012/
3	Visualgo – Interactive Data Structure Visualizations: https://visualgo.net/en
4	GeeksforGeeks – Advanced Data Structures Tutorials: https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-and-algorithms/

Semester: I					
Data Science and Management					
Course Code	**:**	1MCS102	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Basic statistics and probability
Knowledge on Fundamental Concepts	Spreadsheet or basic programming skills
Knowledge on Connected Tools (If any)	Familiarity with data tables and elementary business decision-making.

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the principles of data science, analytics, artificial intelligence, data governance, and their role in data-driven decision making.
CLO2	Apply statistical analysis, visualization, predictive analytics, AI techniques, and business analytics tools to solve real-world data-driven problems.
CLO3	Analyze business problems using predictive, prescriptive, text, web, and social media analytics to support intelligent decision making.
CLO4	Evaluate analytical solutions considering business objectives, ethical issues, privacy, security, responsible AI, and governance principles.
CLO5	Design and develop an end-to-end analytics solution for an industrial application by integrating data science methodologies, AI techniques, governance practices, and business strategy.

Module – 1	**08 Hrs**
Foundations of Data Science and Analytics: Data-Analytic Thinking, Data Science Engineering Data-Driven Decision Making, Business Problems and Data Science Solutions, Data Mining Process, Business Intelligence and Analytics, Analytics Ecosystem, Applications of Analytics.	
Module – 2	**08 Hrs**
Predictive Analytics and Decision Support: Nature of Data, Data Sources, Statistical Analysis, Descriptive Analytics, Data Visualization for Decision Making, Dashboard Concepts, Predictive Modeling, Model Evaluation and Cost-Benefit Analysis.	
Module – 3	**08 Hrs**
Artificial Intelligence and Intelligent Decision Systems: Artificial Intelligence Foundations, AI and Analytics, Intelligent Systems, Text Analytics, Web Analytics, Social Media Analytics, Prescriptive Analytics, Decision Automation and Optimization.	
Module – 4	**08 Hrs**

Data Governance, Ethics and Responsible AI: Big Data Concepts, Big Data Ecosystem, Big Data Applications, Privacy Challenges, Security Concerns, Ethical Issues in Analytics, Responsible AI, Future Challenges in AI Governance.

Module – 5	**08 Hrs**
Data Science Strategy and Industrial Applications: Decision Analysis, Analytic Engineering, Deployment of Analytics, Data Science and Business Strategy, Competitive Advantage through Analytics, Data Science Teams, Organizational Adoption of Analytics, Industrial Case Studies in Healthcare, Manufacturing, Finance, and Transportation.	

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the principles of data science, analytics, artificial intelligence, data governance, and their role in data-driven decision making.
CO2	Apply statistical analysis, visualization, predictive analytics, AI techniques, and business analytics tools to solve real-world data-driven problems.
CO3	Analyze business problems using predictive, prescriptive, text, web, and social media analytics to support intelligent decision making.
CO4	Evaluate analytical solutions considering business objectives, ethical issues, privacy, security, responsible AI, and governance principles.
CO5	Design and develop an end-to-end analytics solution for an industrial application by integrating data science methodologies, AI techniques, governance practices, and business strategy.

Textbooks	
1	Foster Provost and Tom Fawcett, Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking, ,2nd Edition, O'Reilly Media, USA, 2023.

2	Ramesh Sharda, Dursun Delen and Efraim Turban Business Analytics, Data Science and AI: A Managerial Approach, ,6th Edition, Pearson Education, 2026.

Reference Books	
1	Thomas H. Davenport and Jeanne G. Harris, Competing on Analytics: The New Science of Winning, Updated Edition, Harvard Business Review Press, 2017.
2	Thomas H. Davenport, Jeanne G. Harris and Robert Morison, Analytics at Work: Smarter Decisions, Better Results, Harvard Business Review Press.
3	Doug Rose, Artificial Intelligence for Business, FT Press (Pearson), 2018.
4	Bernard Marr, Data Strategy: How to Profit from a World of Big Data, Analytics and Artificial Intelligence, 2nd Edition, Kogan Page, 2022.
5	S. Christian Albright and Wayne L. Winston, Business Analytics: Data Analysis and Decision Making, 8th Edition, Cengage Learning, 2024.

Web links and Video Lectures (e – Resources)	
1	Google Data Analytics: https://grow.google/intl/en_in/data-analytics-course/
2	Elements of AI: https://course.elementsofai.com/?utm_source=chatgpt.com
3	Data Science: https://skillsbuild.org/students/course-catalog/data science?utm_source=chatgpt.com
4	Data Science: https://www.datadays.org/?utm_source=chatgpt.com

Semester: I					
Advanced Database Management Systems					
Course Code	:	1MCS104A	**Credits**	:	3
Hours/Semester (L:T:P:SL)	:	42:0:0:48	**CIE**	:	50 Marks
Total Hours	:	90 Hrs	**SEE**	:	50 Marks
Course Type	:	PCC	**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Fundamentals of databases and the relational model
Knowledge on Fundamental Concepts	Basic data structures and algorithms; introductory computer systems concepts.
Knowledge on Connected Tools (If any)	SQL

Course Learning Objectives: This course will enable the students to	
CLO1	Design advanced relational and object-relational databases with strict integrity constraints.
CLO2	Design and optimize high-performance data architectures by utilizing advanced file Structures, hashing, RAID systems, and distributed database techniques.
CLO3	Implement scalable Big Data solutions using diverse NoSQL architectures, Hadoop Ecosystem, MapReduce frameworks, and distributed storage systems.
CLO4	Build enhanced data models and information retrieval strategies to optimize advanced web search systems.
CLO5	Design data warehouses and apply data mining algorithms to extract actionable business Intelligence patterns.

Module – 1	**08 Hrs**
Review of Relational Data Model and Relational Database Constraints: Relational model concepts; Relational model constraints and relational database schemas; Update operations, dealing with constraint violations. Object and Object-Relational Databases: Overview of Object Database Concepts, Object Database Extensions to SQL, The ODMG Object Model and the Object Definition Language ODL, Object Database Conceptual Design, The Object Query Language OQL, Overview of the C++ Language Binding in the ODMG Standard.	
Module – 2	**08 Hrs**
Disk Storage, Basic File Structures, Hashing, and Modern Storage Architectures: Introduction, Secondary Storage Devices, Buffering of Blocks, Placing File Records on Disk Operations on Files, Files of Unordered Records (Heap Files), Files of Ordered Records (Sorted Files), Hashing Techniques, Other Primary File Organizations, Parallelizing Disk Access Using RAID Technology, Modern Storage Architectures. **Distributed Database Concepts:** Distributed Database Concepts, Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design, Overview of Concurrency Control and Recovery in Distributed Databases, Query Processing and Optimization in Distributed	

Databases, Types of Distributed Database Systems, Distributed Database Architectures, Distributed Catalogue Management.

Module – 3	**08 Hrs**

NOSQL Databases and Big Data Storage Systems: Introduction to NOSQL Systems, The CAP Theorem, Document-Based NOSQL Systems and MongoDB, NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j.

Big Data Technologies Based on MapReduce and Hadoop: What Is Big Data? Introduction to MapReduce and Hadoop, Hadoop Distributed File System (HDFS), MapReduce: Additional Details Hadoop v2 alias YARN, General Discussion.

Module – 4	**08 Hrs**

Enhanced Data Models: Introduction to Active, Temporal, Spatial, Multimedia, and Deductive Databases: Active Database Concepts and Triggers, Temporal Database Concepts, Spatial Database Concepts, Multimedia Database Concepts, Introduction to Deductive Databases.

Introduction to Information Retrieval and Web Search: Information Retrieval (IR) Concepts, Retrieval Models, Types of Queries in IR Systems, Text pre-processing, Inverted Indexing, Evaluation Measures of Search relevance, web Search and Analysis. Trends in Information Retrieval

Module – 5	**08 Hrs**

Data Mining Concepts: Overview of Data Mining Technology, Association Rules, Classification, Clustering, Approaches to Other Data Mining Problems, Applications of Data Mining, Commercial Data Mining Tools
Overview of Data Warehousing and OLAP: Introduction, Definitions, and Terminology, Characteristics of Data Warehouses, Data Modelling for Data Warehouses, building a Data Warehouse, Typical Functionality of a Data Warehouse, Data Warehouse versus Views, Difficulties of Implementing Data Warehouses.

Course Outcomes: After completing the course, the students will be able to	
CO1	Design advanced relational and object-relational databases with strict integrity constraints.
CO2	Design and optimize high-performance data architectures by utilizing advanced file Structures, hashing, RAID systems, and distributed database techniques.
CO3	Implement scalable Big Data solutions using diverse NoSQL architectures, Hadoop Ecosystem, MapReduce frameworks, and distributed storage systems.
CO4	Build enhanced data models and information retrieval strategies to optimize advanced web search systems.
CO5	Design data warehouses and apply data mining algorithms to extract actionable business Intelligence patterns.

Text Books	
1	Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, Pearson Education, 2016.

Reference Books

1	Database System Concepts Author: Abraham Silberschatz, Henry F. Korth, and S. Sudarshan Publisher: McGraw-Hill Year of Publication: March 2022 (7th Edition) ISBN: 9780078022159
2	Advanced Database Management System Author: Dr. Sanjeev Sharma Publisher: Wiley India Year of Publication: 2024 ISBN: 9789351199380.
3	Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, McGraw Hill, 3rd edition, 2013.

Web links and Video Lectures (e – Resources)	
1	https://www.alessandroferrini.com/posts/relational-data-modeling/
2	https://www.geeksforgeeks.org/dbms/definition-and-overview-of-odbms/
3	https://www.mongodb.com/resources/basics/databases/what-is-an-object-oriented database
4	https://www.mongodb.com/resources/basics/databases/distributed-database
5	https://www.geeksforgeeks.org/introduction-to-nosql/
6	https://www.geeksforgeeks.org/dbms/rdms-vs-hadoop/
7	https://www.datacamp.com/tutorial/dbms-data-models-explained
8	https://learn.microsoft.com/en-us/azure/hdinsight/hadoop/apache-hadoop introduction
9	https://www.coveo.com/blog/top-information-retrieval-techniques-and-algorithms/
10	https://peliqan.io/blog/data-warehousing-and-data-mining/
11	https://www.itransition.com/business-intelligence/data-warehousing/olap

Semester: I					
Advances in Computer Networks					
Course Code	**:**	1MCS104B	**Credits**	**:**	3
Hours/Semester (L:T:P:L)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	120 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	IPCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Computer networks fundamentals
Knowledge on Fundamental Concepts	Including Ethernet, IP addressing, TCP/UDP, routing, and basic protocol concepts.
Knowledge on Connected Tools (If any)	Familiarity with network tools is helpful

Course Learning Objectives: This course will enable the students to	
CLO1	Apply link-layer protocols, error detection and correction techniques, multiple access protocols, and data centre networking architectures to ensure reliable and efficient data communication.
CLO2	Select wireless and mobile networking technologies for modern communication systems.
CLO3	Apply multimedia networking protocols to design quality of service aware real-time multimedia communication systems.
CLO4	Utilize network virtualization technologies on multiple operating systems enforcing security enhancements

Module – 1	**08 Hrs**
The Link Layer: Links, Access Neworks and LANs: Introduction to the Link Layer, The Services Provided by the Link Layer, Where Is the Link Layer Implemented? Error **Detection and -Correction Techniques:** Parity Checks, Checksumming Methods, Cyclic Redundancy Check (CRC), **Multiple Access Links and Protocols:** Channel Partitioning Protocols, Random Access Protocols, Taking-Turns Protocols, DOCSIS: The Link-Layer Protocol for Cable Internet Access, **Data Center Networking, Retrospective: A Day in the Life of a Web Page Request:** Getting Started: DHCP, UDP, IP, and Ethernet, Still Getting Started: DNS and ARP, Still Getting Started: Intra-Domain Routing to the DNS Server, Web Client-Server Interaction: TCP and HTTP..	
Module – 2	**08 Hrs**
Wireless and Mobile Networks: Introduction, Wireless Links and Network Characteristics, CDMA,	

Cellular Internet Access: An Overview of Cellular Network Architecture, 3G Cellular Data Networks: Extending the Internet to Cellular Subscribers, On to 4G: LTE,

Mobility Management: Principles: Addressing, Routing to a Mobile Node, **Mobile IP:**

Managing Mobility in Cellular Networks: Routing Calls to a Mobile User, Handoffs in GSM

Module – 3	**08 Hrs**

Multimedia Networking: Multimedia Networking Applications, Properties of Video, Properties of Audio, Types of Multimedia Network Applications, **Streaming Stored Video:** UDP Streaming, HTTP Streaming, Adaptive Streaming and DASH, Content Distribution Networks,
Protocols for Real-Time Conversational Applications: RTP, SIP

Module – 4	**08 Hrs**

Network Virtualization: VPNs, NATs, And Overlays: Introduction, Virtualization, Virtual Private Networks (VPNs), VPN Tunnelling And IP-in-IP Encapsulation, VPN Addressing And Forwarding, Extending VPN Technology To Individual Hosts, Using A VPN With Private IP Addresses, Network Address Translation (NAT), NAT Translation Table Creation, Variant Of NAT, An Example Of NAT Translation, Interaction Between NAT And ICMP, Interaction Between NAT And Applications, NAT In The Presence Of Fragmentation, Conceptual Address Domains, Linux, Windows, And Mac Versions Of NAT, Overlay Networks, Multiple Simultaneous Overlays.

Module – 5	**08 Hrs**

Voice and Video Over IP: Introduction, Digitizing And Encoding, Audio And Video Transmission And Reproduction, Jitter And Playback Delay, Real-time Transport Protocol (RTP), Streams, Mixing, And Multicasting, RTP Encapsulation, RTP Control Protocol (RTCP), RTCP Operation, IP Telephony and Signalling, Quality of Service Controversy, QoS, Utilization, and Capacity, Emergency Services And Pre-emption, IntServ and Resource Reservation, DiffServ and Per-Hop Behaviour, Traffic Scheduling, Traffic Policing And Shaping.

PRACTICAL COMPONENTS OF IPCC

List of experiments

1. Implement Parity Check, Checksum, and Cyclic Redundancy Check (CRC) algorithms for error detection and compare their effectiveness. (**Software Tool:** C/Python)

2. Capture and analyse Ethernet, ARP, IP, TCP, UDP, DNS, and HTTP packets using Wireshark. (**Software Tool:** Wireshark)

3. Simulate Pure ALOHA, Slotted ALOHA, CSMA/CD, and CSMA/CA protocols and compare throughput and collision performance. (**Software Tool:** NS-3 / OMNeT++)

4. Configure an IEEE 802.11 Wireless LAN and evaluate throughput, delay, and packet loss under different traffic loads. (**Software Tool:** Cisco Packet Tracer / NS-3)

5. Configure and analyze Bluetooth and ZigBee communication for short-range wireless

networking. (**Software Tool:** Cisco Packet Tracer / NS-3)

6. Implement Mobile IP concepts and analyze mobility management and handoff performance. (Software Tool: NS-3)

7. Configure VPN and NAT in a virtual network and study packet forwarding and address translation. (**Software Tool:** Cisco Packet Tracer / GNS3 / Linux)

8. Configure audio/video streaming applications and compare UDP streaming with HTTP Adaptive Streaming (DASH). (**Software Tool**: VLC / NS-3)

9. Simulate RTP, RTCP, SIP, and QoS mechanisms and measure jitter, latency, throughput, and packet loss. (**Software Tool:** NS-3 / OMNeT++)

10. Implement Quality of Service (QoS) using DiffServ and evaluate its impact on latency, jitter, throughput, and packet loss. (**Software Tool:** Cisco Packet Tracer, GNS3)

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply link-layer protocols, error detection and correction techniques, multiple access protocols, and data centre networking architectures to ensure reliable and efficient data communication.
CO2	Select wireless and mobile networking technologies for modern communication systems.
CO3	Apply multimedia networking protocols to design quality of service aware real-time multimedia communication systems.
CO4	Utilize network virtualization technologies on multiple operating systems enforcing security enhancements.
CO 5	Apply Voice and Video over IP protocols to ensure Quality of Service (QoS) based traffic management.

Textbooks	
1	James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 6th Edition, Pearson, 2021.
2	Douglas E. Comer, Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture, 6th Edition, Pearson Education, 2014.

Reference Books	
1	Andrew S. Tanenbaum, Nick Feamster, and David J. Wetherall, Computer Networks, 6th Edition, Pearson Education, 2021.
2	Network Warrior – by Gary A. Donahue, Practical guide for real-world network administration.

Web links and Video Lectures (e – Resources)	
1	https://nptel.ac.in/courses/106106243
2	https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs35
3	https://www-net.cs.umass.edu/kurose_ross/index.php
4	https://www.wireshark.org/docs/wsug_html_chunked/

Semester: I					
Advances in Operating System					
Course Code	:	1MCS104C	**Credits**	:	3
Hours/Week (L:T:P)	:	42:0:48	**CIE**	:	50 Marks
Total Hours	:	90	**SEE**	:	50 Marks
Course Type	:	IPCC	**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Operating-system fundamentals
Knowledge on Fundamental Concepts	Processes and threads, CPU scheduling, synchronization
Knowledge on Connected Tools (If any)	Memory management, and basic networking/programming

Course Learning Objectives: This course will enable the students to	
CLO1	Apply the principles of real-time operating systems, scheduling algorithms, resource sharing mechanisms, and the different types of distributed systems.
CLO2	Analyse and compare centralized, decentralized, and hybrid distributed system architectures, multithreading, virtualization, server clusters, and code migration techniques for distributed applications.
CLO3	Apply communication protocols, synchronization algorithms, logical clocks, mutual exclusion techniques, and election algorithms to solve distributed computing problems.
CLO4	Evaluate fault tolerance mechanisms, process resilience, reliable communication protocols, distributed commit protocols, and recovery techniques to design dependable distributed systems.
CLO5	Build secure distributed applications by incorporating authentication, authorization, secure channels, security management, and distributed file system architectures

Module – 1	**08Hrs**
Introduction: Introduction to Operating Systems, Goals and Functions of an Operating System, Multiprocessor and Multicore Systems, Clustered Systems, Operating System Services, Resource Management, Computing Environments, Open-Source Operating Systems, Operating System Design Principles, user and Kernel Modes, System Calls.	
Module – 2	**08Hrs**
Architectures: Architectural Styles, Centralized Architectures, Decentralized Architectures, Hybrid Architectures, Threads: Introduction to Threads, Threads in Distributed Systems, The Role of Virtualization in Distributed Systems, Architectures of Virtual Machines, Networked User Interfaces, Client-Side Software for Distribution Transparency. Servers: General Design Issues, Server Clusters, Managing Server Clusters, Approaches to Code Migration, Migration and Local Resources	
Module – 3	**08Hrs**
Communication: Fundamentals, Remote Procedure Call, Message-Oriented Communication, Stream-Oriented Communication, Multicast Communication, Synchronization: Clock Synchronization, Logical Clocks, Mutual Exclusion, Global Positioning of Nodes, Election Algorithms.	

Module – 4	**08Hrs**
Fault Tolerance: Introduction To Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit, Recovery.	
Module – 5	**10Hrs**
Security: Introduction To Security, Secure Channels, Access Control, Security Management, Distributed File Systems, Architecture, Processes, Communication, Synchronization, Consistency and Replication.	

PRACTICAL COMPONENTS OF IPCC

List of experiments

1. Study and configure Linux Real-Time Scheduling Policies (SCHED_FIFO and SCHED_RR), Simulate Earliest Deadline First (EDF) Scheduling.
2. Implement the Liu & Layland CPU Utilization Test for periodic real-time tasks and determine whether a task set is schedulable under Rate Monotonic Scheduling (RMS). Implement Priority Inheritance Protocol using Semaphores.
3. Design and implement a GUI-based distributed chat application that enables multiple users to communicate simultaneously over a network using a client-server architecture.
4. Design and implement a simple TCP Client-Server application using socket programming, where the client sends a message to the server and the server responds with an acknowledgment.
5. Implement an RPC-based distributed application that enables a client to invoke procedures on a remote server transparently, without explicitly handling network communication.
6. Implement the Berkeley Clock Synchronization Algorithm for synchronizing clocks of distributed systems.
7. Simulate the Vector Clock Algorithm for maintaining the causal ordering of events in a distributed system.
8. Implement a multithreaded TCP server using POSIX threads (pthread) that can handle multiple client connections simultaneously.
9. Simulate a server cluster and implement Round Robin Load Balancing for distributing client requests among multiple servers. Develop Transparent Client Interface for Distributed Application.
10. Develop a server-cluster simulation consisting of one load balancer and at least three backend servers. Implement the following load balancing

algorithms for reduced response time.

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply the principles of real-time operating systems, scheduling algorithms, resource sharing mechanisms, and the different types of distributed systems.
CO2	Analyse and compare centralized, decentralized, and hybrid distributed system architectures, multithreading, virtualization, server clusters, and code migration techniques for distributed applications.
CO3	Apply communication protocols, synchronization algorithms, logical clocks, mutual exclusion techniques, and election algorithms to solve distributed computing problems.
CO4	Evaluate fault tolerance mechanisms, process resilience, reliable communication protocols, distributed commit protocols, and recovery techniques to design dependable distributed systems.
CO5	Build secure distributed applications by incorporating authentication, authorization, secure channels, security management, and distributed file system architectures.

Text Books	
1	A. Silberschatz, Peter B Galvin, Greg Gagne - Operating System Concepts, Wiley India, 10th Edition, 2023 (ISBN: 978-9357460569)
2	Andrew Tanenbaum Maarten Van Steen - Distributed Systems Principles and Paradigms, 2nd Edition, Pearson, Thirteenth Impression 2025.

Reference Books	
1	Mukesh Singhal and Niranjan G. Shivaratri, *Advanced Concepts in Operating Systems,* McGraw -Hill, 1st Edition, 2014.
2	George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, *Distributed Systems: Concepts and Design*, 5th Edition, Pearson, 12th Impression, 2026.
3	Andrew S. Tanenbaum and Herbert Bos, *Modern Operating Systems*, 5th Edition, Pearson, 2024.

Web links and Video Lectures (e – Resources)	
1	https://www.youtube.com/playlist?list=PLWkTsO24LpD8ufLfZz2t3e4LSI05ueIOq **:** NPTEL videos for Operating System**.**
2	https://www.youtube.com/watch?v=NzZXz3fJf6o&list=PLShJJCRzJWxhz7SfG4hpaB D5bKOloWx9J **:** NPTEL Link for cloud computing overview.

Semester: I					
Advanced Bigdata Analytics					
Course Code	**:**	**1MCS104D**	**Credits**	**:**	4
Hours/Week(L:T:P)	**:**	42: 0:28 :50	**CIE**	**:**	50 Marks
Total Hours	**:**	120 Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	IPCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Programming fundamentals (preferably Python);
Knowledge on Fundamental Concepts	Basic statistics and probability; introductory data structures;
Knowledge on Connected Tools (If any)	Familiarity with data tables is useful

Course Learning Objectives: This course will enable the students to	
CLO1	Describe the need of Big data and visualization in real world applications.
CLO2	Apply big data analytics frameworks and visualization techniques for decision making
CLO3	Analyse the use of big data analytics and visualization for business applications
CLO4	Formulate optimized relational and scripting-based queries using Apache Hive and Pig to handle large-scale data ETL and reporting operations

Module – 1	**08 Hrs**
Introduction to Big Data Analytics: Definition, Need of Big Data and its characteristics, Classification of Data - Structured, semi structured, and unstructured, Scalability and Parallel Processing, Designing Data Architecture, Data Sources, Quality, Pre-Processing and Storing, Data Storage and Analysis, Big Data Analytics Applications and Case Studies	
Module – 2	**08 Hrs**
Introduction to Hadoop: Introduction, Hadoop and its Ecosystem, Hadoop Distributed File System, MapReduce Framework and Programming Model, Hadoop Yarn, Hadoop Ecosystem Tools, Hadoop Distributed File System Basics: HDFS Design Features, Components, HDFS User Commands.	
Module – 3	**08 Hrs**
MapReduce, Hive and Pig: Introduction, MapReduce Map Tasks, Reduce Tasks and MapReduce Execution, Composing MapReduce for Calculations and Algorithms, Hive, HiveQL, Pig	
Module – 4	**08 Hrs**
Spark and Big Data Analytics: Introduction to Data Analysis with Spark-Spark SQL,	

Using Python with Spark SQL, Data Analysis Operations, Data ETL process, Introduction Analytics, Reporting and Visualization.	
Module – 5	**08 Hrs**
Machine Learning Algorithms for Big Data Analytics: Introduction, Estimating the relationships, Outliers, Variances, Probability Distributions, and Correlations, Regression analysis, Finding Similar Items, Similarity of Sets and Collaborative Filtering, Clustering Analysis.	

Course Outcomes

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

Course Outcomes: After completing the course, the students will be able to	
CO1	Describe the need of Big data and visualization in real world applications.
CO2	Apply big data analytics frameworks and visualization techniques for decision making
CO3	Analyse the use of big data analytics and visualization for business applications
CO4	Formulate optimized relational and scripting-based queries using Apache Hive and Pig to handle large-scale data ETL and reporting operations

Textbooks	
1	Raj Kamal and Preeti Saxena, "Big Data Analytics Introduction to Hadoop, Spark, and Machine-Learning", McGraw Hill Education, 2018 ISBN: 9789353164966, 9353164966.
2	Douglas Eadline, "Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem", 1st Edition, Pearson Education, 2016. ISBN-13: 978- 9332570351

Reference Books	
1	Tom White, "Hadoop: The Definitive Guide", 4th Edition, O"Reilly Media, 2015.ISBN-13: 978- 9352130672.
2	Seema Acharya, Subhashini Chellappan, Big Data and analytics, Wiley Publications, 2015, ISBN-10: 8126554789, ISBN-13: 978-8126554782.

Web links and Video Lectures (e – Resources)	
1	Essential for mastering HDFS commands, YARN configurations, and core MapReduce programming models: https://hadoop.apache.org/docs/stable/
2	Architecture scalability, Hadoop ecosystem tools, Pig, and Hive: https://nptel.ac.in/courses/106105186

Semester: I					
Artificial Intelligence for IoT					
Course Code	**:**	1MCS105A	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Basic programming and mathematics
Knowledge on Fundamental Concepts	Introductory statistics and probability
Knowledge on Connected Tools (If any)	Elementary machine-learning concepts are helpful but not assumed

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the fundamental concepts of Artificial Intelligence, Internet of Things (IoT), and their integration for developing intelligent IoT systems.
CLO2	Apply machine learning, deep learning, and reinforcement learning techniques for processing and analyzing IoT-generated data.
CLO3	Develop AI-enabled IoT applications using cloud platforms and intelligent data analytics techniques
CLO4	Analyze AI techniques for processing visual, text, audio, speech, and time-series data generated by IoT devices.
CLO5	Apply AIoT concepts to develop solutions for smart homes, healthcare, smart cities, and industrial automation.

Module – 1	**08 Hrs**
Foundations and Basic Integrations of IoT and AI **Principles and Foundations of IoT and AI**: What is IoT 101? Big data and IoT, Infusion of AI data science in IoT, Tools used. **Machine Learning for IoT:** Understanding ML and its Role in IoT, Learning paradigms, Prediction using linear regression, Logistic regression for classification, Classification using support vector machines, Naive Bayes, Decision trees, Ensemble learning.	
Module – 2	**08 Hrs**

Advanced AI Techniques and Their Application in IoT

Deep Learning for IoT: Deep learning 101, Multilayered perceptron for regression and classification, Understanding CNNs, Recurrent neural networks.

Reinforcement Learning for IoT: Introduction to RL, Simulated environments, Introducing Gymnasium, Q-learning, Q-Network

Module – 3	**08 Hrs**

Implementing Intelligent IOT Solutions in Diverse Domains-1

AI Cloud Platforms for IoT: Introduction to Cloud Computing for IoT, Cloud Computing Basics, AI Services in Cloud Platforms, Training and Deploying AI Models for IoT on Cloud Platforms

Deep Learning for Time-Series Data from IoT: Introduction to Time-Series Data, The Importance of Time-Series Analysis in IoT,

Module – 4	**08 Hrs**

Implementing Intelligent IOT Solutions in Diverse Domains-2

Leveraging AI for Visual Data from IoT: A Brief Introduction to Visual Data in IoT Ecosystems, Handling Challenges in Visual Data in IoT, Leveraging AI Techniques for Visual Data in IoT.

AI for Text, Audio and Speech Data from IoT: Introduction to Text, Audio, and Speech Data, Fundamentals of Text Data Processing, Working with Audio Data in IoT.

Module – 5	**08 Hrs**

Applying AI and IoT in Real-World Scenarios

AI for Personal and Home IoT: Wearable and Fitness Devices, Smartwatch Health Monitoring in Action, Smart Home Devices, AI-driven home automation and intelligence, Case study: An Integrated smart home Implementation

AI for Smart Cities IoT:

AI Applications in Smart Cities: Smart traffic management, Smart parking, Smart waste management, smart policing. AI-Powered Urban Analytics, Adapting IoT for Smart Cities and the Necessary Steps.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the fundamental concepts of Artificial Intelligence, Internet of Things (IoT), and their integration for developing intelligent IoT systems.
CO2	Apply machine learning, deep learning, and reinforcement learning techniques for processing and analyzing IoT-generated data.
CO3	Develop AI-enabled IoT applications using cloud platforms and intelligent data analytics techniques

CO4	Analyze AI techniques for processing visual, text, audio, speech, and time-series data generated by IoT devices.
CO5	Apply AIoT concepts to develop solutions for smart homes, healthcare, smart cities, and industrial automation.

Textbooks	
1	Dr. Amita Kapoor, *Hands-On Artificial Intelligence for IoT*, 2nd Edition, May 2025, Packt Publishing Ltd., Birmingham, UK, ISBN: 978-1-83546-718-3. www.packtpub.com

Reference Books	
1	Yuxi (Hayden) Liu, *Artificial Intelligence for IoT Cookbook*, 1st Edition, March 2021, Packt Publishing Ltd., Birmingham, UK, ISBN: 978-1-83898-198-5. www.packtpub.com
2	Rajkumar Buyya and Amir Vahid Dastjerdi (Editors), *Internet of Things: Principles and Paradigms*, 1st Edition, May 2016, Morgan Kaufmann (Elsevier), Cambridge, Massachusetts, USA, ISBN: 978-0-12-805395-9
3	Perry Lea, *Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security*, 1st Edition, January 2018, Packt Publishing Ltd., Birmingham, UK, ISBN: 978-1-78847-059-9 (Paperback). www.packtpub.com

Web links and Video Lectures (e – Resources)	
1	NPTEL – Introduction to Internet of Things: https://nptel.ac.in
2	NPTEL – Artificial Intelligence and Machine Learning: https://nptel.ac.in
3	Google Cloud AI and IoT: https://cloud.google.com
4	Coursera – AI and Machine Learning Courses: https://www.coursera.org
5	edX – Artificial Intelligence and IoT Courses: https://www.edx.org
6	Artificial Intelligence for IoT: https://www.udacity.com/course/aiot-foundations--ud074?utm_source=chatgpt.com

Semester: I					
Artificial Intelligence for Cyber Security					
Course Code	**:**	1MCS105B	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50Marks
Total Hours	**:**	90 Hrs	**SEE**	**:**	50Marks
Course Type	**:**	Elective	**SEE Duration**	**:**	3hr

Prerequisites:	
Courses	Basic programming (Python recommended.)
Knowledge on Fundamental Concepts	Linear algebra, probability, statistics, and introductory machine learning;
Knowledge on Connected Tools (If any)	Familiarity with images is useful

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the fundamental concepts of AI and ML and their applications in addressing cybersecurity challenges.
CLO2	Apply AI and ML techniques to detect, analyze, and mitigate cyber security threats and vulnerabilities
CLO3	Utilize Natural Language Processing (NLP) methods in cybersecurity applications and analyze strategies for defending against adversarial attacks on AI systems.
CLO4	Evaluate the ethical, legal, and societal implications of deploying AI technologies in cybersecurity environments

Module – 1	**08 Hrs**
Why Machine Learning and Security? : Cyber Threat Landscape, The Cyber Attacker's Economy , A Marketplace for Hacking Skills, Indirect Monetization, , What Machine Learning Is Not, Adversaries Using Machine Learning, Real-World Uses of Machine Learning in Security , Spam Fighting: An Iterative Approach and Limitations of Machine Learning in Security. **Classifying and Clustering:** Machine Learning: Problems and Approaches, Machine Learning in Practice: A Worked Example Training Algorithms to Learn.	
Module – 2	**08 Hrs**
Classifying and Clustering: Supervised Classification Algorithms, Practical Considerations in Classification Clustering. **Anomaly Detection :** When to Use Anomaly Detection Versus Supervised Learning**,** Intrusion Detection with Heuristics, Data-Driven Methods, Feature Engineering for Anomaly Detection, Anomaly Detection with Data and Algorithms, Practical System Design Concerns , Malware Analysis.	
Module – 3	**08 Hrs**

Network Traffic Analysis: Theory of Network Defense**,** Machine Learning and Network Security, Building a Predictive Model to Classify Network Attacks , Natural Language Processing in Cybersecurity.

Protecting the Consumer Web: Types of Abuse and the Data That Can Stop Them, Authentication and Account Takeover, Supervised Learning for Abuse Problems , Clustering Abuse, Further Directions in Clustering.

Module – 4	**08 Hrs**

Production Systems: Defining Machine Learning System Maturity and Scalability, Data Quality**,** Model Quality**,** Performance**,** Maintainability**,** Monitoring and Alerting **and** Security and Reliability.

Module – 5	**08 Hrs**

Adversarial Machine Learning: The Importance of Adversarial ML, Security Vulnerabilities in Machine Learning Algorithms, Attack Transferability, Attack Technique: Model Poisoning, Attack Technique: Evasion Attack.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the fundamental concepts of AI and ML and their applications in addressing cybersecurity challenges.
CO2	Apply AI and ML techniques to detect, analyze, and mitigate cyber security threats and vulnerabilities
CO3	Utilize Natural Language Processing (NLP) methods in cybersecurity applications and analyze strategies for defending against adversarial attacks on AI systems.
CO4	Evaluate the ethical, legal, and societal implications of deploying AI technologies in cybersecurity environments

Text Books	
1	Clarence Chio and David Freeman, Machine Learning and Security: Protecting Systems with Data and Algorithms, 2018, O'REILLY.

Reference Books	
1	David Kim, Michael G. Solomon, —Fundamentals of Information Systems Security, Jones & Bartlett Learning Publishers, 2013.
2	Nina Godbole, Sunit Belapure, —Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives‖, Wiley Publishers, 2011.
3	"AI in Cybersecurity: Transforming Threat Detection and Response" by Arun Kumar Krishna, 2024.;
4	Artificial Intelligence in Cyber Security" by Leslie F. Sikos, Springer, 2019.

Web links and Video Lectures (e – Resources)	
1	Google Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course *Comprehensive introduction to machine learning concepts, algorithms, and practical applications.*
2	NPTEL – Introduction to Machine Learning: https://nptel.ac.in/courses/106106139 *Covers supervised learning, classification, clustering, and model evaluation.*
3	MIT OpenCourseWare – Introduction to Machine Learning: https://ocw.mit.edu/courses/6-390-introduction-to-machine-learning-fall-2024/
4	Stanford CS224N – Natural Language Processing with Deep Learning: https://web.stanford.edu/class/cs224n/ *Useful for the NLP in Cybersecurity component of the course.*
5	Coursera – Machine Learning for Cyber Threat Detection: https://www.coursera.org/learn/machine-learning-cyber-threat-and-anomaly-detection-mqcsa *Focuses on anomaly detection, malware analysis, and cybersecurity applications of machine learning*

Semester: I					
Advanced Computer Vision					
Course Code	**:**	1MCS105C	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Basic statistics and probability; data handling and visualization
Knowledge on Fundamental Concepts	Introductory data mining or machine learning;
Knowledge on Connected Tools (If any)	Spreadsheet/Python skills are helpful.

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the basic concepts of computer vision, image formation, and digital cameras.
CLO2	Apply image processing, filtering, and optimization techniques to analyze images.
CLO3	Use deep learning models such as neural networks and CNNs for image recognition tasks.
CLO4	Apply camera geometry and calibration techniques for 3D vision applications.
CLO5	Analyze transformation models and multi-view vision techniques for computer vision problems.

Module – 1	**08 Hrs**
Introduction: What is computer vision? **Image Formation:** Geometric primitives and transformation, Photometric image formation, Digital Camera.	
Module – 2	**08 Hrs**
Image Processing: Point operators, Linear filtering, Neighborhood operators, Fourier Transform. **Model Fitting and Optimization:** Scattered Data Interpolation, Variational Methods and Regularization.	
Module – 3	**08 Hrs**
Deep Learning: Supervised Learning, Unsupervised Learning, Deep Neural Networks, Convolution Neural Networks (CNNs), Training Deep Networks. **Recognition:** Instance Recognition, Image Classification, Object Detection, Semantic Segmentation.	
Module – 4	**08 Hrs**

Image Preprocessing and Feature Extraction, The Pinhole Camera, Three Geometric Problems, Homogeneous Coordinates, Learning Extrinsic Parameters, Learning Intrinsic Parameters, Inferring Three-Dimensional World Points, Applications of the Pinhole Camera.
Module – 5 **08 Hrs**
Transformation Models and Multi-View Vision: Two-Dimensional Transformation Models, Learning in Transformation Models, Inference in Transformation Models, Multiple Cameras, Models for Shape, Models for Style and Identity, Temporal Models, Models for Visual

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the basic concepts of computer vision, image formation, and digital cameras.
CO2	Apply image processing, filtering, and optimization techniques to analyze images.
CO3	Use deep learning models such as neural networks and CNNs for image recognition tasks.
CO4	Apply camera geometry and calibration techniques for 3D vision applications.
CO5	Analyze transformation models and multi-view vision techniques for computer vision problems.

Textbooks	
1	Computer Vision: Algorithms and Applications – Richard Szeliski, 2nd Edition, Springer, 2022
2	Computer Vision: Models, Learning, and Inference – Simon J. D. Prince, 1st Edition, Cambridge University Press,2012

Reference Books	
1	Aston Zhang et al., Dive into Deep Learning, Latest Online Edition, 2023–2025
2	Kevin P. Murphy, Probabilistic Machine Learning: An Introduction, 2022**.**

Web links and Video Lectures (e – Resources)	
1	Computer Vision: Algorithms and Applications (2nd Edition, 2022) – Online: https://szeliski.org/Book/
2	Dive into Deep Learning – https://d2l.ai/
3	Neural Networks and Deep Learning – http://neuralnetworksanddeeplearning.com/
4	Stanford CS231n Course – http://cs231n.stanford.edu/
5	MIT OpenCourseWare – Computer Vision – https://ocw.mit.edu/
6	Coursera Computer Vision (University of Michigan) – https://www.coursera.org/
7	OpenCV Documentation – https://docs.opencv.org/
8	PyTorch Tutorials – https://pytorch.org/tutorials/
9	TensorFlow Tutorials – https://www.tensorflow.org/tutorials

Semester: I					
Data Analytics for Fraud Detectio					
Course Code	**:**	1MSCS105D	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Programming fundamentals; software development life cycle;.
Knowledge on Fundamental Concepts	Object-oriented concepts;
Knowledge on Connected Tools (If any)	Basic web technologies (HTML/CSS/JavaScript) and databases

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the concepts, types, and challenges of fraud detection using data analytics.
CLO2	Apply data preprocessing, visualization, and analytical techniques for fraud analysis.
CLO3	Develop fraud detection models using statistical and machine learning techniques.
CLO4	Evaluate fraud detection solutions using predictive analytics, graph analytics, and real-world case studies.

Module – 1	**08 Hrs**
Introduction to Fraud Analytics: Fundamentals and characteristics of fraud, types of fraud, fraud detection and prevention, role of big data in fraud detection, data-driven fraud detection, fraud-detection techniques, fraud cycle, fraud analytics process model, role of fraud data scientists, data-analytic thinking, business problems and data science solutions, supervised and unsupervised analytical approaches, and overview of the data mining process.	
Module – 2	**08 Hrs**
Data Collection, Sampling and Preprocessing: Types of data sources, merging data sources, sampling, types of data elements, visual data exploration and exploratory statistical analysis, Benford's Law, descriptive statistics, treatment of missing values and outliers, identification of red flags, data standardization and categorization, weights of evidence coding, variable selection, Principal Component Analysis, RIDIT and PRIDIT analysis, and segmentation.	
Module – 3	**08 Hrs**

Descriptive Analytics and Outlier Detection: Descriptive analytics for fraud detection, fundamentals of anomaly and outlier detection, graphical and statistical outlier detection procedures, probabilistic and statistical models for outlier detection, linear models, proximity-based outlier detection, clustering techniques for fraud detection, high-dimensional outlier detection, and one-class Support Vector Machines for identifying anomalous observations.

Module – 4	**08 Hrs**

Predictive Analytics for Fraud Detection: Fundamentals of predictive modelling, target definition, supervised learning, logistic regression, variable selection, decision trees, neural networks, Support Vector Machines, ensemble methods, multiclass classification, model generalization and overfitting, model validation, evaluation of classification models, handling skewed and imbalanced datasets, and performance evaluation of fraud detection models

Module – 5	**08 Hrs**

Social Network Analytics, Fraud Model Management and Governance: Social networks and their application to fraud detection, fraud as a social phenomenon, homophily, neighbourhood-based metrics, community mining for identifying groups of fraudsters, bipartite networks, fraud model lifecycle, model representation, selection of cases for investigation, fraud alert and case management, visual analytics, backtesting analytical fraud models, model design and documentation, data quality, privacy in fraud analytics, economic considerations, modelling extensions, Internet of Things in fraud analytics, and corporate fraud governance.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the concepts, types, and challenges of fraud detection using data analytics.
CO2	Apply data preprocessing, visualization, and analytical techniques for fraud analysis.
CO3	Develop fraud detection models using statistical and machine learning techniques.
CO4	Evaluate fraud detection solutions using predictive analytics, graph analytics, and real-world case studies.

Textbooks	
1	Bart Baesens, Véronique Van Vlasselaer and Wouter Verbeke, *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection*, John Wiley & Sons, 1st Edition, 2015.
2	Foster Provost and Tom Fawcett, *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*, O'Reilly Media, 2013.ISBN 9781449374273.
3	Charu C. Aggarwal, *Outlier Analysis*, 2nd Edition, Springer, 2017

Reference Books	
1	Jiawei Han, Micheline Kamber, and Jian Pei, Data Mining: Concepts and Techniques, 4th Edition, Morgan Kaufmann, 2023.
2	Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, 2nd Edition (Corrected), Springer.
3	Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th Edition, Morgan Kaufmann, 2016.
4	Christopher M. Bishop and Hugh Bishop, Deep Learning: Foundations and Concepts, Springer, 2024.
5	Kevin Murphy, Probabilistic Machine Learning: An Introduction, MIT Press, 2022

Web links and Video Lectures (e – Resources)	
1	Kaggle Datasets (Fraud Detection) – Public datasets for credit card fraud, insurance fraud, transaction analytics, and machine learning projects.
2	Scikit-learn Documentation – Documentation and tutorials for machine learning algorithms used in fraud detection.
3	Scikit-learn MOOC (Inria) – Free hands-on course covering data preprocessing, classification, anomaly detection, and model evaluation.

Semester: I					
Software Engineering and Web Applications					
Course Code	**:**	**1MCS106A**	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Basic computer networks and distributed systems concepts;
Knowledge on Fundamental Concepts	Discrete mathematics
Knowledge on Connected Tools (If any)	Elementary cryptography and programming are helpful

Course Learning Objectives: This course will enable the students to	
CLO1	Apply software architecture, component-based development, and service-oriented principles in designing large-scale software systems.
CLO2	Analyse software reliability, security, quality, and dependability using appropriate engineering techniques and metrics.
CLO3	Apply software evolution, configuration management, and process improvement practices to enhance software maintainability.
CLO4	Apply web engineering principles to design effective web application architecture, navigation, and user interfaces.
CLO5	Apply web engineering practices to develop, test, deploy, and maintain secure and high-quality web applications.

Module – 1	**08 Hrs**
Software Architecture and Reuse Engineering: Architectural Design – Architectural Design Decisions, Architectural Views, Architectural Patterns, Application Architectures , Component-Based Software Engineering – Components and Component Models, CBSE Process, Component Composition, Service-Oriented Software Engineering – Service-Oriented Architecture (SOA), RESTful Services, Service engineering, Service Composition.	
Module – 2	**08 Hrs**
Software Dependability and Security: Dependable systems: Dependability Properties, Sociotechnical systems, Redundancy and diversity, Dependable processes, Reliability Engineering – Availability and Reliability, Reliability Requirements, Fault-Tolerant Architectures, Reliability Measurement, Security Engineering – Security and dependability Security Requirements, Secure Systems Design, Security Testing and Assurance.	
Module – 3	**08 Hrs**

Processes, Legacy Systems, Software Maintenance, Quality Management – Software Quality, Quality Assurance, Software Standards, Reviews and Inspections, Software Measurement. Configuration Management – Version Management, System Building, Change Management, Release Management
Module – 4 **08 Hrs**
Web Application Analysis and Design: Introduction to Web Applications, Characteristics of Web Applications, Web Application Analysis, Content Design, Architectural Design for Web Applications, Navigation Design, Interface Design, User Experience Design, Responsive Web Design Principles, Modeling Web Applications.
Module – 5 **08 Hrs**
Web Application Development, Testing and Maintenance: Web Application Construction, Web Components, Web Services Integration, Testing Web Applications, Functional Testing, Performance Testing, Quality Assurance for Web Applications, Web Application Security, Deployment Strategies, Web Application Maintenance and Evolution.

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply software architecture, component-based development, and service-oriented principles in designing large-scale software systems.
CO2	Analyse software reliability, security, quality, and dependability using appropriate engineering techniques and metrics.
CO3	Apply software evolution, configuration management, and process improvement practices to enhance software maintainability.
CO4	Apply web engineering principles to design effective web application architecture, navigation, and user interfaces.
CO5	Apply web engineering practices to develop, test, deploy, and maintain secure and high-quality web applications.

Text Books	
1	Software Engineering, Ian Sommerville, Pearson Education, 10th Edition, 2016.
2	Web Engineering A Practitioner's Approach Roger S. Pressman and David Lowe, McGraw-Hill Higher Education, 1st Edition, 2008.

Reference Books	
1	Software Architecture in Practice Len Bass, Paul Clements, and Rick Kazman, Addison-Wesley Professional, 4th Edition, 2021.
2	Web Engineering Gerti Kappel, Birgit Pröll, Siegfried Reich, andWerner Retschitzegger, Wiley, 2006.
3	DevOps A Software Architect's Perspective Len Bass, Ingo Weber, and Liming Zhu, Addison-Wesley Professional, 2015.

Web links and Video Lectures (e – Resources)	
1	https://ocw.mit.edu/search/?q=software+engineering
2	https://developer.mozilla.org/en-US/docs/Learn_web_development
3	https://www.w3schools.com/

Semester: I						
Advances in Blockchain Technology						
Course Code	:	**1MCS106B**		Credits	:	03
Hours/Week (L:T:P:SL)	:	42:0:0:48		CIE	:	50 Marks
Total Hours	:	90 Hrs		SEE	:	50 Marks
Course Type	:	PCC		SEE Duration	:	3 Hours

Prerequisites:	
Courses	Programming fundamentals
Knowledge on Fundamental Concepts	Web technologies and HTTP; basic networking, databases, and security concepts
Knowledge on Connected Tools (If any)	Familiarity with mobile applications is useful.

Course Learning Objectives: This course will enable the students to	
CLO1	Analyze blockchain architecture, decentralization principles, and cryptographic foundations.
CLO2	Evaluate advanced consensus protocols and blockchain scalability solutions.
CLO3	Implement smart contracts and decentralized applications using Ethereum
CLO4	Compare enterprise blockchain frameworks and their application domains.
CLO5	Assess blockchain security, privacy-preserving mechanisms, and emerging decentralized technologies

Module – 1	**08 Hrs**
Blockchain Fundamentals and Distributed Ledger Technology: Introduction to Blockchain, Distributed Systems, CAP Theorem and FLP Impossibility, Decentralization using Blockchain, Types of Blockchains, Consensus Mechanisms, Cryptographic Primitives, Hash Functions, Public Key Cryptography, Digital Signatures, Merkle Trees.	
Module – 2	**08 Hrs**
Advanced Consensus and Scalability Mechanisms: Byzantine Generals Problem, Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Authority (PoA), Federated Consensus, Blockchain Scalability Challenges, Segregated Witness (SegWit), State Channels, Lightning Network, Sidechains.	
Module – 3	**08 Hrs**
Ethereum, Smart Contracts and Decentralized Applications: Ethereum Architecture, Ethereum Virtual Machine (EVM), Accounts and Transactions, Gas Mechanism, Smart Contracts, Solidity Fundamentals, Smart Contract Deployment, Decentralized Applications (DApps), Decentralized Autonomous Organizations (DAOs), Token Standards.	

Module – 4	08 Hrs
Enterprise Blockchain Platforms: Enterprise Blockchain Concepts, Hyperledger Project, Hyperledger Fabric Architecture, Membership Service Provider, Channels and Chaincode, Hyperledger Sawtooth, Hyperledger Iroha, Corda Architecture, Quorum Platform, Enterprise Use Cases.	
Module – 5	**08 Hrs**
Security, Privacy and Emerging Trends: Blockchain Security, 51% Attacks, Sybil Attacks, Eclipse Attacks, Privacy in Blockchain, Zero-Knowledge Proofs, Ring Signatures, Confidential Transactions, Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), Web3 Ecosystem, Blockchain and IoT, Blockchain and Artificial Intelligence.	

Course Outcomes: After completing the course, the students will be able to	
CO1	Analyze blockchain architecture, decentralization principles, and cryptographic foundations.
CO2	Evaluate advanced consensus protocols and blockchain scalability solutions.
CO3	Implement smart contracts and decentralized applications using Ethereum
CO4	Compare enterprise blockchain frameworks and their application domains.
CO5	Assess blockchain security, privacy-preserving mechanisms, and emerging decentralized technologies.

Text Books	
1	Imran Bashir, *Mastering Blockchain*, 4th Edition, Packt Publishing, 2023.

Reference Books	
1	Andreas M. Antonopoulos and Gavin Wood, *Mastering Ethereum: Building Smart Contracts and DApps*, 1st Edition, O'Reilly Media, Inc., USA, 2018. ISBN: 978-1491971949.
2	Daniel Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 2nd Edition, Apress, Berkeley, CA, USA, 2020. ISBN: 978-1484254127.
3	Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, 1st Edition, Princeton University Press, Princeton, New Jersey, USA, 2016. ISBN: 978-0691171692.

Web links and Video Lectures (e – Resources)	
1	Ethereum: https://ethereum.org/en/developers/docs
2	Hyperledger: https://hyperledger-fabric.readthedocs.io
3	Blockchain Basics: https://www.ibm.com/topics/blockchain Blockchain Foundation Training: https://training.linuxfoundation.org/blockchain

Semester: I						
Mobile and Web Application Security						
Course Code	:	**1MCS106C**		**Credits**	:	03
Hours/Week(L:T:P)	:	42: 0: 0: 48		**CIE**	:	50 Marks
Total Hours	:	90 Hrs		**SEE**	:	50 Marks
Course Type	:	PEC		**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Programming (Python/SQL recommended); l
Knowledge on Fundamental Concepts	Relational databases and SQL; basic statistics;
Knowledge on Connected Tools (If any)	Familiarity with cloud and data-processing concepts is helpfu

Course Learning Objectives: This course will enable the students to	
CLO1	Describe principles of secure design and development for web and mobile applications.
CLO2	Analyze vulnerabilities and attacks affecting modern application ecosystems.
CLO3	Apply security mechanisms for authentication, authorization, data protection, and secure communications.
CLO4	Perform security testing and vulnerability assessment of applications.
CLO5	Investigate DevSecOps practices and emerging trends in application security.

Module – 1	**08 Hrs**
Secure Application Engineering and Threat Modeling Introduction to application security; security goals and principles; attack surfaces and trust boundaries; threat modeling methodologies (STRIDE, DREAD); secure software development life cycle (SSDLC); risk assessment; security requirements engineering; OWASP Web Top-10 and Mobile Top-10; secure coding principles.	
Module – 2	**08 Hrs**
Web Application Security HTTP/HTTPS security; authentication and authorization mechanisms; session management; cookies and token security; Same-Origin Policy; Cross-Site Scripting (XSS); Cross-Site Request Forgery (CSRF); SQL Injection; file upload vulnerabilities; REST API security; Content Security Policy and security headers.	
Module – 3	**08 Hrs**

Mobile Application Security

Android application architecture; permission model; secure data storage; cryptographic APIs; secure inter-process communication; secure networking and TLS; code signing; reverse engineering; obfuscation techniques; malware and privacy threats; mobile application testing.

Module – 4	08 Hrs

Application reconnaissance and mapping; vulnerability discovery; authentication attacks; session attacks; access-control vulnerabilities; Static Application Security Testing (SAST); Dynamic Application Security Testing (DAST); vulnerability scanning; secure code review; penetration testing methodologies; remediation and reporting.

Module – 5	08 Hrs

DevSecOps and Emerging Trends

DevSecOps principles; CI/CD security; secure containerized deployments; microservices and API ecosystem security; cloud-native application security; privacy engineering; AI-assisted security tools; zero-trust principles; case studies of major web and mobile security breaches; current research challenges.

Course Outcomes: After completing the course, the students will be able to	
CO1	Describe principles of secure design and development for web and mobile applications.
CO2	Analyze vulnerabilities and attacks affecting modern application ecosystems.
CO3	Apply security mechanisms for authentication, authorization, data protection, and secure communications.
CO4	Perform security testing and vulnerability assessment of applications.
CO5	Investigate DevSecOps practices and emerging trends in application security.

Text Books	
1	Andrew Hoffman, Web Application Security: Exploitation and Countermeasures for Modern Web Applications, 2nd Edition, O'Reilly Media (Indian Edition available through Shroff Publishers), 2024.
2	Pragati Ogal Rai, Shubham Asthana, Android Application Security Essentials, Packt Publishing, 2015.

Reference Books	
1	Dafydd Stuttard and Marcus Pinto, The Web Application Hacker's Handbook, , Wiley, 2011.

2	Joshua J. Drake, Zach Lanier, Collin Mulliner, Pau Oliva Fora, Stephen A. Ridley, Georg Wicherski , Android Hacker's Handbook, Wiley, 2014.
3	OWASP Web Security Testing Guide and Mobile Application Security Testing Guide (latest editions, freely available online).

Web links and Video Lectures (e – Resources)	
1	OWASP Foundation, https://owasp.org , OWASP Top -10, Secure Coding, Testing Guides
2	https://owasp.org/www-project-web-security-testing-guide/, Web Application Security Testing
3	https://owasp.org/www-project-mobile-app-security/ : Mobile Security Standards
4	https://www.youtube.com/@OWASPGlobal OWASP Foundation YouTube Channel

Semester: I						
Data Engineering						
Course Code	:	**1MCS106D**		**Credits**	:	03
Hours/Week(L:T:P)	:	42: 0: 0: 48		**CIE**	:	50 Marks
Total Hours	:	90 Hrs		**SEE**	:	50 Marks
Course Type	:	PEC		**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Programming fundamentals;
Knowledge on Fundamental Concepts	Software testing concepts; basic web applications and APIs;
Knowledge on Connected Tools (If any)	Familiarity with JavaScript/TypeScript and command-line tools is helpful

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the concepts of data engineering, data architecture, data engineering lifecycle, and the roles and responsibilities of data engineers in modern organizations.
CLO2	Apply data engineering principles to select appropriate technologies, architectures, and lifecycle components for developing scalable, reliable, and maintainable data platforms.
CLO3	Analyze data generation, storage, ingestion, transformation, and serving techniques to support efficient data processing and analytics requirements.
CLO4	Evaluate data engineering solutions by considering data quality, scalability, security, privacy, governance, technology selection, and operational requirements in modern data ecosystems.
CLO5	Evaluate modern data engineering tools, platforms, and emerging technologies to recommend appropriate solutions for diverse data engineering applications and organizational requirements.

Module – 1	**08 Hrs**
Foundations of Data Engineering and Data Architecture: Data Engineering Defined, The Data Engineering Lifecycle, Evolution of the Data Engineer, Data Engineering and Data Science, Data Engineering Skills and Activities, Data Maturity and the Data Engineer, Business Responsibilities, Technical Responsibilities, Data Engineers Inside an Organization, Internal-Facing versus External-Facing Data Engineers, Data Engineering Lifecycle Components, Designing Good Data Architecture, Principles of Good Data Architecture, Scalability, Security, Loosely Coupled Systems, Reversible Decisions, Architecture Leadership.	
Module – 2	**08 Hrs**
Data Engineering Lifecycle and Technology Selection: The Data Engineering Lifecycle, Source Systems, Storage, Ingestion, Transformation, Serving Data, Security, Data Management, DataOps, Orchestration, Software Engineering, Choosing Technologies Across the Data Engineering Lifecycle, Evaluating Technology Choices, Build versus Buy Decisions, Cloud and On-Premises Considerations, Total Cost of Ownership, Vendor Selection and Technology Strategy	
Module – 3	**08 Hrs**

Data Generation, Storage, and Ingestion: Data Generation in Source Systems, APIs, Application Databases, OLTP and OLAP Systems, Change Data Capture, Logs, Message Queues and Event Streams, Data Sharing, Storage Fundamentals, File Storage, Block Storage, Object Storage, Distributed Storage Systems, Data Warehouses, Data Lakes, Data Lakehouses, Data Platforms, Data Ingestion, Batch and Stream Ingestion, ETL and ELT, Serialization and Deserialization, Throughput, Reliability, Scalability, Data Migration.

Module – 4	**08 Hrs**
Queries, Transformation, and Data Serving: Queries, Data Modeling, Normalization and Denormalization, Data Transformation, Data Integration, Data Quality, Data Pipelines, Analytics Engineering, Serving Data for Analytics, Business Analytics, Operational Analytics, Embedded Analytics, Machine Learning Support, Data Products, Data Mesh, Query Federation, Semantic and Metrics Layers, Reverse ETL, Data Sharing and Self-Service Analytics.	

Module – 5	**08 Hrs**
Security, Privacy, and Future Trends in Data Engineering: Security Fundamentals, Security Policies, Principle of Least Privilege, Encryption, Logging, Monitoring and Alerting, Network Access, Data Privacy, Governance Considerations, Shared Responsibility in Cloud Environments, Data Backup and Recovery, Emerging Data Engineering Trends, Modern Data Stack, Live Data Stack, Streaming Pipelines, Real-Time Analytics, Data and Machine Learning Integration, Future Roles and Responsibilities of Data Engineers.	

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the concepts of data engineering, data architecture, data engineering lifecycle, and the roles and responsibilities of data engineers in modern organizations.
CO2	Apply data engineering principles to select appropriate technologies, architectures, and lifecycle components for developing scalable, reliable, and maintainable data platforms.
CO3	Analyze data generation, storage, ingestion, transformation, and serving techniques to support efficient data processing and analytics requirements.
CO4	Evaluate data engineering solutions by considering data quality, scalability, security, privacy, governance, technology selection, and operational requirements in modern data ecosystems.
CO5	Evaluate modern data engineering tools, platforms, and emerging technologies to recommend appropriate solutions for diverse data engineering applications and organizational requirements.

Text Books	
1	Joe Reis and Matt Housley, Fundamentals of Data Engineering: Plan and Build Robust Data Systems, O'Reilly Media, 2022.

Reference Books	
1	Martin Kleppmann and Chris Riccomini, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, 2nd Edition, O'Reilly Media, 2026.
2	James Densmore, Data Pipelines Pocket Reference: Moving and Processing Data for Analytics, 1st Edition, O'Reilly Media, 2021.
3	Tyler Akidau, Slava Chernyak, and Reuven Lax, Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing, 1st Edition, O'Reilly Media, 2018.
4	Ralph Kimball and Margy Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition, John Wiley & Sons, 2013.

Web links and Video Lectures (e – Resources)	
1	Data Engineering Zoomcamp: https://www.youtube.com/@DataTalksClub/playlists
2	Data Engineering Zoom camp GitHub Repository: https://github.com/DataTalksClub/data-engineering-zoomcamp
3	Datatalks.club: https://www.youtube.com/@DataTalksClub?utm_source=chatgpt.com

Semester: I						
Software Automation Testing Laboratory						
Course Code	:	**1MCSL107A**		**Credits**	:	01
Hours/Week(L:T:P)	:	0: 0: 28 :2		**CIE**	:	50 Marks
Total Hours	:	30 Hrs		**SEE**	:	50 Marks
Course Type	:	PCC		**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Programming with Python;
Knowledge on Fundamental Concepts	Basic linear algebra and image representation;
Knowledge on Connected Tools (If any)	Familiarity with NumPy/OpenCV is helpful.

Course Learning Objectives: This course will enable the students to	
CLO1	Implement automated UI and API test scripts using Playwright and WireMock to validate software functionality.
CLO2	Analyze automation test results, identify defects, evaluate test coverage, and troubleshoot failures using appropriate testing tools and reports.
CLO3	Evaluate automation frameworks, testing strategies, and test artifacts to improve software quality and testing efficiency.
CLO4	Develop a scalable end-to-end automation testing framework by integrating UI testing, API mocking, reporting, and continuous testing practices for real-world applications.

PART – A
List of Experiments

1. Develop and execute an automated test using Playwright to validate the login functionality of a web application with valid and invalid credentials. Analyze the execution results to identify failed scenarios and evaluate the effectiveness of the implemented automation script in improving software quality.
2. Design and implement automated tests using Playwright to validate the search and filtering functionality of an e-commerce application with multiple input combinations. Analyze the test results for different scenarios and evaluate the robustness and maintainability of the automation scripts.
3. Develop an end-to-end automated test suite using Playwright to validate shopping cart and checkout functionalities of an e-commerce application. Analyze the execution outcomes and evaluate the test suite based on coverage, reliability, and execution efficiency.
4. Configure WireMock to simulate a REST API using predefined stub responses for GET requests. Analyze the mocked API responses under different request conditions and evaluate the effectiveness of service virtualization in

supporting application testing.

5. Create WireMock stubs to simulate POST requests with valid and invalid input data for a REST API. Analyze the application's behavior using mocked responses and evaluate the suitability of WireMock for testing applications when backend services are unavailable.
6. Integrate Playwright with WireMock to automate the testing of a web application that interacts with mocked REST APIs. Analyze the interaction between the user interface and mocked services and evaluate the reliability, repeatability, and efficiency of the integrated automation testing approach.

PART – B

OPEN ENDED EXPERIMENTS

Design and Implement an End-to-End Automated Testing Framework for a Web Application: Develop a scalable automated testing framework for any open-source web application (banking, education, healthcare, etc.) using Playwright. Identify critical user workflows, automate the test scenarios, execute them under different conditions, and analyze the test outcomes.

1. **Design and implement a data-driven automated testing framework using Playwright /** Selenium / Available open source **for a selected web application.** Identify critical workflows and execute the same test scenarios using multiple sets of valid, invalid, boundary, and unexpected input data. Implement parameterization, test fixtures, reusable functions, and appropriate assertions. Analyze the test results to identify defects and evaluate the maintainability and reusability of the framework.
2. **Develop an automated testing solution using Playwright /** Selenium / Available open source **to evaluate a selected web application across multiple browsers and viewport configurations.** Identify critical functional and UI scenarios, execute them across different browser engines and screen sizes, capture failures and screenshots, and compare the test results. Analyze browser-specific and responsive behaviour and evaluate the effectiveness of automated cross-browser testing.
4. **Design and implement a service virtualization framework using WireMock for a selected REST-based application.** Identify different API response conditions such as successful responses, validation errors, authentication failures, timeouts, unavailable services, and server errors. Create suitable WireMock stubs and test how the application responds to these conditions. Analyze the results and evaluate the effectiveness of API contract and negative testing**Develop an automated testing framework using Playwright /** Selenium / Available open source **that generates comprehensive test execution reports for a selected web application.** Execute a suitable set of functional test cases, capture screenshots and traces for failed scenarios, categorize failures, and analyze test execution time, pass/fail statistics, and coverage. Develop a mechanism to identify recurring failures and evaluate the effectiveness of automated reporting in

software quality assessment.

5. **Design and implement a Continuous Testing pipeline for a web application by integrating Playwright-based automated tests with a CI/CD platform such as GitHub Actions or Jenkins.** Configure automated test execution whenever application changes are introduced, collect test results and artifacts, and configure the pipeline to identify and report failures. Analyze the benefits and limitations of continuous testing in improving software quality and reducing regression testing effort.
6. **Design and implement a scalable end-to-end quality engineering framework for a selected open-source web application by integrating UI automation using Playwright, API simulation using WireMock, reusable test components, test data management, automated reporting, and continuous test execution.** Identify critical business workflows, develop functional and negative test scenarios, execute the tests under different conditions, analyze failures and coverage, and evaluate the scalability, reliability, maintainability, and efficiency of the proposed framework

Course Outcomes: After completing the course, the students will be able to	
CO1	Implement automated UI and API test scripts using Playwright and WireMock to validate software functionality.
CO2	Analyze automation test results, identify defects, evaluate test coverage, and troubleshoot failures using appropriate testing tools and reports.
CO3	Evaluate automation frameworks, testing strategies, and test artifacts to improve software quality and testing efficiency.
CO4	Develop a scalable end-to-end automation testing framework by integrating UI testing, API mocking, reporting, and continuous testing practices for real-world applications.

Text Books	
1	Paul C. Jorgensen, The Craft of Software Testing: Managing the Testing Process, 3rd Edition, Auerbach Publications.
2	Alan Richardson, *Selenium WebDriver Recipes in C#, 2nd Edition*, Leanpub
3	Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing: Principles and Practices*, Pearson Education

Reference Books	
1	Evan Burchard, The Web Application Hacker's Handbook, *Wiley*..

Web links and Video Lectures (e – Resources)	
1	https://playwright.dev
2	https://www.wiremock.io/webinars/developer-portals
3	https://playwright.dev/community/learn-videos

Semester: I						
Computer Vision Laboratory						
Course Code	**:**	**1MCSL107B**		**Credits**	**:**	01
Hours/Week(L:T:P)	**:**	0: 0: 28 :2		**CIE**	**:**	50 Marks
Total Hours	**:**	30 Hrs		**SEE**	**:**	50 Marks
Course Type	**:**	PCC		**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Programming in C/C++/Java/Python
Knowledge on Fundamental Concepts	Core data structures and algorithms
Knowledge on Connected Tools (If any)	Basic algorithmic complexity and debugging skills

Course Learning Objectives: This course will enable the students to	
CLO1	Apply basic image processing techniques such as gray scale conversion, filtering, histogram analysis, and segmentation
CLO2	Implement edge, corner, and feature detection methods to extract important information from images.
CLO3	Apply feature matching techniques to compare and analyze images.
CLO4	Develop and evaluate basic deep learning models (CNN) for image classification tasks.
CLO5	Design and implement computer vision solutions for real-world applications such as object detection, face recognition, and motion analysis

PART – A
List of Experiments

1. Write a program to read an image, convert it into grayscale, display its histogram, and perform geometric transformations such as translation, rotation, and scaling.
2. Write a program to apply smoothing filters (Average and Gaussian) and detect edges using Sobel and Canny edge detection methods.
3. Write a program to detect corner points using Harris Corner Detection and perform feature matching between two images using ORB descriptors.
4. Write a program to segment an image using thresholding techniques and analyze the segmented regions.
5. Write a program to build and train a Convolutional Neural Network (CNN) for image classification using a standard dataset such as MNIST or CIFAR-10.

6. Write a program to estimate camera intrinsic and extrinsic parameters using
calibration images and perform basic 3D point reconstruction or depth estimation.

PART – B

OPEN ENDED EXPERIMENTS

Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course
coordinator. It encourages creativity, critical thinking, and inquiry-based learning.

1. Design and develop a Real-Time Object Detection System using computer vision and deep learning techniques for detecting and localizing objects in images or videos.
2. Design and develop a Face Recognition or Face Mask Detection System using feature extraction and deep learning techniques.
3. Design and develop an Image Segmentation System for medical, satellite, or industrial images and evaluate its performance.
4. Design and develop an Image Stitching (Panorama Creation) System using feature detection, matching, and geometric transformations.
5. Design and develop an Optical Flow-Based Motion Detection and Tracking System for video analysis applications.
6. Design and develop an Image Classification System using Transfer Learning with
pre-trained deep learning models and benchmark datasets.

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply basic image processing techniques such as gray scale conversion, filtering, histogram analysis, and segmentation
CO2	Implement edge, corner, and feature detection methods to extract important information from images.
CO3	Apply feature matching techniques to compare and analyze images.
CO4	Develop and evaluate basic deep learning models (CNN) for image classification tasks.
CO5	Design and implement computer vision solutions for real-world applications such as object detection, face recognition, and motion analysis

Text Books	
1	Computer Vision: Algorithms and Applications – Richard Szeliski, 2nd Edition, Springer, 2022.
2	Computer Vision: Models, Learning, and Inference – Simon J. D. Prince, 1st Edition, Cambridge University Press, 2012.
3	Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 2016.

Reference Books	
1	Dive into Deep Learning – Aston Zhang et al., Latest Online Edition (2023–2025).
2	Probabilistic Machine Learning: An Introduction – Kevin P. Murphy, MIT Press, 2022.
3	Learning OpenCV 4 Computer Vision with Python – Joseph Howse, Joe Minichino, Packt, 2020.

Web links and Video Lectures (e – Resources)	
1	Computer Vision: Algorithms and Applications (Online Book)
2	Dive into Deep Learning (Interactive Book)
3	MIT OpenCourseWare – Computer Vision

Semester: I						
Advanced Data Structures Laboratory						
Course Code	:	**1MCSL107C**		**Credits**	:	01
Hours/Week(L:T:P)	:	0: 0: 28 :2		**CIE**	:	50 Marks
Total Hours	:	30 Hrs		**SEE**	:	50 Marks
Course Type	:	PCC		**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Programming in Java
Knowledge on Fundamental Concepts	Object-oriented programming; basic HTTP/REST and JSON; introductory databases and SQL
Knowledge on Connected Tools (If any)	Familiarity with Maven/Gradle is helpful.

Course Learning Objectives: This course will enable the students to	
CLO1	Make use of appropriate advanced data structures to solve complex, real-world computational problems
CLO2	Develop code to perform data operations and validate results.
CLO3	Make use of appropriate programming tools and development environments to implement, test, and debug advanced data structure applications.
CLO4	Build a technical document consisting of experiment setup, design, implementation, and results with inferences.

PART – A
List of Experiments

1. Task Priority Selection System (Heap)

   - Insert tasks with priorities
   - Identify and execute highest priority task

2. Job Queue Merging System (Leftist Tree)
   - Merge two job queues
   - Maintain priority ordering after merge

3. Adaptive Search System (Splay Tree)

   - Insert and search elements
   - Frequently accessed elements move to root

4. Structured Data Indexing (B-Tree)

   - Insert keys into structured index
   - Maintain sorted hierarchical organization

5. Dynamic Key Storage System (Extendible Hashing)

- Insert records into hash buckets
- Handle overflow using directory expansion

6.Social Group Formation System (DSU)

- Perform union operations
- Check connectivity between members

PART – B
OPEN ENDED EXPERIMENTS

Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course coordinator. It encourages creativity, critical thinking, and inquiry-based learning.

1. Concept 1: Stock Monitoring System (Heap)
   Track minimum and maximum stock values dynamically
2. Concept 2: Event List Combination System (Leftist Tree)
   Merge multiple event lists into one ordered list
3. Concept 3: Recently Used Item Tracker (Splay Tree)
   Track frequently accessed elements using self-adjusting behavior
4. Concept 4: Search Index Verification System (B-Tree)
   Search for existence of keys in structured index
5. Concept 5: Bucket Distribution System (Extendible Hashing)
   Observe record distribution across dynamically growing buckets
6. Concept 6: Network Connectivity Checker (DSU)
   Determine whether two nodes belong to the same connected group

Course Outcomes: After completing the course, the students will be able to	
CO1	Make use of appropriate advanced data structures to solve complex, real-world computational problems
CO2	Develop code to perform data operations and validate results.
CO3	Make use of appropriate programming tools and development environments to implement, test, and debug advanced data structure applications.
CO4	Build a technical document consisting of experiment setup, design, implementation, and results with inferences.

Text Books	
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press.
2	Ellis Horowitz and Sartaj Sahni, *Fundamentals of Data Structures in C*, 2nd Edition (2023), Universities Press(India) Private limited.

Reference Books	
1	Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition, (2014) Pearson Education.
2	Sartaj Sahni, Data Structures, Algorithms and Applications in C++, 2nd Edition, (2011), Universities Press.
3	Peter Brass, Advanced Data Structures, 1st Edition, (2011), Cambridge University Press.

Web links and Video Lectures (e – Resources)	
1	NPTEL – Advanced Data Structures: https://nptel.ac.in/courses/106102064
2	MIT OpenCourseWare – Advanced Data Structures: https://ocw.mit.edu/courses/6-851-advanced-data-structures-spring-2012/
3	VisuAlgo – Data Structure Visualization: https://visualgo.net/en

Semester: I						
Micro Services Development Laboratory						
Course Code	:	**1MCSL107D**		**Credits**	:	01
Hours/Week(L:T:P)	:	0: 0: 28 :2		**CIE**	:	50 Marks
Total Hours	:	30 Hrs		**SEE**	:	50 Marks
Course Type	:	PCC		**SEE Duration**	:	3 Hours

Prerequisites:	
Courses	Basic programming;
Knowledge on Fundamental Concepts	Object-oriented programming and Java fundamentals;
Knowledge on Connected Tools (If any)	HTTP/REST APIs, JSON, SQL and database persistence; familiarity with Spring Boot and testing tools is useful

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the architecture and components of a Spring Boot microservice application and configure the development environment.
CLO2	Develop RESTful microservices with database connectivity, validation, and exception handling
CLO3	Implement communication, service discovery, and request routing among microservices.
CLO4	Deploy, document, and test microservices using containerization and API testing tools.
CLO5	Develop a simple distributed microservices application by integrating multiple microservices.

PART – A
List of Experiments

PART – A
List of experiments

Experiment 1

Development of a Product Management Microservice

Objective To develop a RESTful Product Management microservice by extending the reusable Spring Boot microservice framework with database persistence, validation, and exception handling.

Description Extend the reusable Spring Boot microservice framework developed during the pre-requisite exercises to implement a Product Management microservice. Design the Product entity and corresponding repository, service, controller, and DTO classes following the layered architecture.

Implement RESTful CRUD operations for managing product information using Spring Data JPA and a relational database. Incorporate input validation and

global exception handling to ensure reliable and maintainable service implementation.

Test the REST APIs using Postman and verify the correctness of all CRUD operations and error-handling mechanisms.

Students shall be able to
Extend an existing Spring Boot microservice framework.
Design entity and DTO classes.
Implement repository, service, and controller layers.
Develop RESTful CRUD APIs.
Configure database persistence using Spring Data JPA.
Apply input validation.
Implement global exception handling.
Test REST APIs using Postman.

Deliverables

Product Management microservice.
Database schema and entity classes.
Functional CRUD REST APIs.
Validated request handling.
Tested API collection.

Experiment 2

Development of a Customer Management Microservice

Objective : To develop an independent Customer Management microservice using the reusable Spring Boot framework and implement RESTful services for customer information management.

Description: Develop a Customer Management microservice by extending the reusable framework created during the pre-requisite exercises. Design the Customer entity together with the corresponding repository, service, controller, and DTO components using the standard layered architecture.

Implement RESTful CRUD APIs for customer management with persistent storage using Spring Data JPA. Apply validation and exception-handling mechanisms to ensure reliable processing of user requests. Verify all APIs using Postman and evaluate the correctness of the implemented microservice.

Students shall be able to

Develop an independent microservice. Design
layered application components. Implement
CRUD services.
Manage persistent customer data.
Apply validation techniques.
Handle application exceptions.
Test REST APIs.

Deliverables

Customer Management microservice.
Functional REST APIs.
Persistent customer database.

Validated API responses.
Tested Postman collection.

Experiment 3
Development of Communicating Microservices

Objective: To develop two independent microservices and implement synchronous inter-service communication using REST-based service invocation mechanisms.

Description: Develop two lightweight microservices by extending the reusable Spring Boot framework. Design suitable REST APIs for each microservice and implement synchronous communication using Spring REST Client or OpenFeign to exchange information between them.

Verify the correctness of service interactions by invoking APIs and exchanging JSON data. Implement appropriate exception handling to manage communication failures and validate the responses using Postman.

Students shall be able to
Develop multiple independent microservices.
Implement synchronous inter-service communication.
Exchange data between services.
Handle communication exceptions.
Test service interactions using Postman.
Deliverables
Two independent microservices.
Functional REST-based service communication.
Successful data exchange between services.
Tested Postman collection.

Experiment 4
Development of Service Discovery and API Gateway

Objective :To configure service registration, service discovery, and centralized API routing in a distributed microservices environment.

Description: Develop two simple microservices using the reusable framework and configure them to register with a Service Discovery Server. Configure an API Gateway to route client requests dynamically to the registered services.
Verify successful service registration, service discovery, and gateway-based request routing. Test different request scenarios using Postman and evaluate the behaviour of the distributed application.

Students shall be able to
Configure a Service Discovery Server.
Register microservices dynamically.
Configure API Gateway routing.
Route client requests through the gateway.
Verify service registration and discovery.
Deliverables

Service Discovery Server. Registered microservices. Configured API Gateway. Successful gateway-based routing.

Experiment 5

Containerization and Deployment of Microservices.

Objective: To containerize and deploy multiple microservices using Docker and Docker Compose.

Description: Develop one or more microservices using the reusable framework and containerize them by creating Docker images and Dockerfiles. Configure Docker Compose to deploy the microservices together with the required database services.

Verify successful deployment, container networking, and application execution in the containerized environment. Evaluate the portability and deployment characteristics of the application.

Students shall be able to

Create Dockerfiles.
Build Docker images.
Configure Docker Compose.
Deploy multiple containers.
Verify container networking and execution.

Deliverables

Dockerfiles. Docker images.
Docker Compose configuration.
Running containerized microservices.

Experiment 6

API Documentation, Functional Testing, and Performance Evaluation.

Objective: To document, test, and evaluate RESTful microservices using API documentation and software testing tools.

Description: Develop a microservice using the reusable framework and generate API documentation using Swagger/OpenAPI. Perform functional testing using Postman by executing various request scenarios and validating the correctness of API responses.

Conduct basic performance evaluation using JMeter or an equivalent testing tool by measuring response time and throughput under different workloads. Analyse the observed performance characteristics and document the test results.

Students shall be able to Generate API documentation. Perform functional API testing. Design API test cases.
Conduct basic performance evaluation.
Analyse application performance.

Deliverables
Swagger/OpenAPI documentation.
Functional API test cases.
Performance testing report. Performance analysis results.

PART – B
OPEN ENDED EXPERIMENTS

Experiment 1
Design and Development of an E-Commerce Microservices Application

Objective: To design and develop a distributed e-commerce application using multiple microservices and contemporary microservices technologies.

Description: Design and develop an e-commerce application by identifying suitable business services such as Product Management, Customer Management, Order Management, Inventory Management, and Payment Processing. Develop the required microservices using the reusable Spring Boot framework and implement REST-based communication among them.
Integrate service discovery, API gateway, database persistence, and containerized deployment. Test the application using Postman and generate API documentation to validate the correctness of the distributed system.

Students shall be able to
Design a distributed microservices architecture.
Identify service boundaries.
Develop multiple collaborating microservices.
Integrate service discovery and API gateway. Deploy
and validate the complete application.

Deliverables
Complete e-commerce microservices application. Integrated distributed services.
API documentation.
Containerized deployment.
Functional testing report.

Experiment 2
Design and Development of a Smart Healthcare Management System

Objective : To design and develop a distributed healthcare management application using multiple collaborating microservices.

Description:Design and develop a healthcare management system by identifying services such as Patient Management, Doctor Management, Appointment Scheduling, Billing, and Medical Records. Develop the required microservices using the reusable framework and implement communication among services.

Configure service discovery, API gateway, database persistence, and deployment using Docker Compose. Test and document the developed application.

Students shall be able to

Model healthcare services as microservices. Develop integrated service-oriented applications. Configure distributed application infrastructure. Test and document the developed solution.

Deliverables

Healthcare microservices application.
Integrated services.
API documentation.
Deployment configuration.
Test report.

Experiment 3

Design and Development of a Secure Microservices Application

Objective : To design and develop a secure microservices application by implementing authentication and authorization mechanisms.

Description: Design and develop a distributed microservices application by integrating authentication and authorization using Spring Security and JSON Web Token (JWT) or equivalent security mechanisms. Protect REST APIs, manage user authentication, and authorize access to secured resources.
Test secured APIs using Postman and verify authentication, authorization, and protected resource access under different user scenarios.

Students shall be able to

Design secure RESTful services.
Implement authentication mechanisms.
Configure authorization rules.
Secure distributed microservices.
Test secured APIs.

Deliverables

Secure microservices application.
Authentication and authorization implementation.
Protected REST APIs.
Security testing report.

Experiment 4

Design and Development of a Cloud-Native Microservices Solution

Objective: To design, develop, deploy, and evaluate a complete cloud-native microservices solution for a real-world application.
Description: Identify a suitable real-world problem and design a cloud-native microservices architecture by decomposing the application into independent services. Develop the required microservices using the reusable framework and integrate service communication, service discovery, API gateway, containerization, API documentation, and testing.
Deploy the complete application using Docker Compose, perform functional testing, and evaluate the overall behaviour, scalability, and maintainability of the developed solution.

Students shall be able to

Design complete cloud-native applications.
Develop integrated distributed systems.

Apply microservices architectural principles.
Deploy containerized applications.
Evaluate the developed solution.
Deliverables
Complete cloud-native application.
Distributed microservices architecture.
Containerized deployment.
API documentation.
Functional testing report.
Project report.

Experiment 5
Design and Development of a Smart Campus Microservices Application

Objective: To design and develop a distributed smart-campus application using independently deployable microservices.

Description: Design and develop a smart-campus system by identifying appropriate services such as Student Management, Attendance, Classroom/Laboratory Management, Asset Management, Notification, and Maintenance Request services. Develop suitable microservices and integrate them using REST-based or event-driven communication. Incorporate database persistence, service discovery, API gateway, security where appropriate, and containerized deployment. Test and document the complete system.
Students shall be able to: Identify microservice boundaries from a real-world problem; design collaborating services; integrate multiple microservices; configure distributed infrastructure; deploy and validate a complete application.

Deliverables: Smart-campus microservices application; architecture diagram; integrated services; API documentation; containerized deployment; testing report.

Experiment 6
Design and Development of a Library Management Microservices Application

Objective: To design and develop a simple Library Management application using multiple microservices.

Description: Design and develop a Library Management application consisting of suitable services such as Book Management, Member Management, and Issue/Return Management. Develop the required microservices using Spring Boot and implement REST-based communication among them. Configure database persistence and test the integrated application using Postman.

Students shall be able to:
Identify services in a simple application.
Develop independent Spring Boot microservices. Implement REST APIs for different services. Establish communication between microservices. Store and retrieve application data. Test the integrated application using Postman.

Deliverables:
Library Management microservices application.

Minimum three functional microservices. REST APIs for the identified services. Database configuration. Postman test collection. Brief architecture diagram.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the architecture and components of a Spring Boot microservice application and configure the development environment.
CO2	Develop RESTful microservices with database connectivity, validation, and exception handling
CO3	Implement communication, service discovery, and request routing among microservices.
CO4	Deploy, document, and test microservices using containerization and API testing tools.
CO5	Develop a simple distributed microservices application by integrating multiple microservices.

Text Books	
1	Chris Richardson, Microservices Patterns: With Examples in Java, Manning Publications, 2018.
2	Sam Newman, Building Microservices: Designing Fine-Grained Systems, 2nd Edition, O'Reilly Media, 2021.
3	Laurentiu Spilca, Spring Start Here: Learn What You Need and Learn It Well, Manning Publications, 2021.

Reference Books	
1	Craig Walls, Spring Boot in Action, Manning Publications, 2016.
2	Mark Richards and Neal Ford, Fundamentals of Software Architecture: An Engineering Approach, O'Reilly Media, 2020.
3	Eberhard Wolff, Microservices: Flexible Software Architecture, Addison-Wesley Professional, 2017.
4	Brendan Burns, Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services, O'Reilly Media, 2018.
5	Spring Boot Reference Documentation (Official).
6	Spring Cloud Reference Documentation (Official).
7	Docker Documentation (Official).

Web links and Video Lectures (e – Resources)

1	Spring Boot Official Documentation
2	Spring Boot Project Home and Getting Started Guides
3	Spring Boot Docker Guide
4	Docker Java Language Guide

Semester: II					
Deep Learning and its Applications					
Course Code	**:**	1MSCS201	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Machine Learning, Artificial Intelligence, Probability and Statistics, Linear Algebra and Python Programming.
Knowledge on Fundamental Concepts	Neural networks, machine learning algorithms, optimization, matrix operations and Python programming.
Knowledge on Connected Tools (If any)	Python, Jupyter Notebook/Google Colab, TensorFlow/Keras and PyTorch.

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental concepts, architectures and learning principles of deep neural networks.
CLO2	Analyze and implement convolutional neural networks, recurrent neural networks and other deep learning architectures.
CLO3	Apply optimization, regularization and transfer learning techniques to improve deep learning models.
CLO4	Evaluate deep learning models using suitable performance measures and select appropriate architectures for different applications.
CLO5	Develop deep learning solutions for computer vision, natural language processing and other real-world applications

Module – 1	**08 Hrs**
Introduction to Deep Learning Introduction to deep learning, biological and artificial neurons, perceptron, multilayer perceptron, feedforward neural networks, activation functions, loss functions, gradient descent, backpropagation, computational graphs, optimization algorithms, vanishing and exploding gradients.	
Module – 2	**08 Hrs**
Convolutional Neural Networks Convolution operation, filters and feature maps, padding and stride, pooling, CNN architecture, training of CNNs, data augmentation, batch normalization, dropout, receptive field, popular CNN architectures: LeNet, AlexNet, VGG, GoogLeNet/Inception, ResNet and MobileNet..	
Module – 3	**08 Hrs**

Sequence Models and Recurrent Neural Networks	
Sequential data and sequence modelling, recurrent neural networks, unfolded computational graphs, BPTT, vanishing and exploding gradients, LSTM, GRU, bidirectional RNNs, encoder-decoder architecture, attention mechanism, sequence classification and sequence generation.	
Module – 4	**08 Hrs**
Advanced Deep Learning Techniques	
Autoencoders, denoising and sparse autoencoders, variational autoencoders, generative adversarial networks, transfer learning, fine-tuning, self-supervised learning concepts, hyperparameter tuning, learning-rate schedules, early stopping and model regularization.	
Module – 5	**08 Hrs**
Applications of Deep Learning	
Deep learning applications in computer vision, image classification, object detection and segmentation, medical image analysis, natural language processing, sentiment analysis, speech and audio processing, recommendation systems, anomaly detection, time-series prediction, explainable AI and ethical considerations	

Topics for Self learning	
Module – 1	Derivation of backpropagation, optimization variants, batch normalization and practical neural-network implementation.
Module – 2	Transfer learning with pretrained CNNs, visualization of feature maps, object detection architectures and image augmentation.
Module – 3	Attention mechanism, sequence-to-sequence learning, beam search and practical LSTM/GRU applications.
Module – 4	VAE and GAN implementation, hyperparameter optimization, transfer learning and self-supervised learning case studies.
Module – 5	Deep learning case studies in healthcare, autonomous systems, NLP, recommender systems and explainable AI.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the principles of deep neural networks, activation functions, loss functions, optimization and backpropagation.
CO2	Design and implement CNN-based models for image classification and computer vision tasks.
CO3	Analyze and apply RNN, LSTM, GRU and attention-based models for sequential data.
CO4	Apply advanced techniques such as autoencoders, GANs, transfer learning and regularization to practical problems.
CO5	Evaluate and develop deep learning applications in computer vision, NLP, healthcare and other real-world domains

Textbooks

1	Deep Learning", Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016.
2	"Deep Learning with Python", François Chollet, 2nd Edition, Manning Publications, 2021

Reference Books	
1	"Dive into Deep Learning", Aston Zhang, Zachary C. Lipton, Mu Li and Alexander J. Smola, Cambridge University Press, 2023.
2	"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", Aurélien Géron, 3rd Edition, O'Reilly Media, 2022.
3	"Pattern Recognition and Machine Learning", Christopher M. Bishop, Springer, 2006.

Web links and Video Lectures (e – Resources)	
1	https://www.deeplearningbook.org/
2	https://www.tensorflow.org/tutorials
3	https://pytorch.org/tutorials/
4	https://www.deeplearning.ai/
5	https://d2l.ai/

Semester: II					
Cryptography and Network Security					
Course Code	**:**	1MSCS202	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Computer Networks, Computer Programming and Discrete Mathematical Structures.
Knowledge on Fundamental Concepts	Encryption/Decryption, cryptographic algorithms, key management, and network security protocols.
Knowledge on Connected Tools (If any)	CISCO Packet Tracer, NS2/NS3

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental concepts of cryptography, including symmetric and asymmetric encryption, and their applications in securing data.
CLO2	Analyze and evaluate various network security protocols, such as SSL/TLS and IPsec, to understand their roles in maintaining data confidentiality and integrity.
CLO3	Implement cryptographic techniques, including hashing and digital signatures, to ensure data authenticity and integrity in software applications.
CLO4	Identify common security threats and vulnerabilities in network systems and propose effective countermeasures to mitigate these risks.
CLO5	Assess real-world security solutions and practices, critically evaluating their effectiveness in addressing contemporary cybersecurity challenges.

Module – 1	**08 Hrs**
Introduction to Cryptography Basics of Cryptography, History and importance, Symmetric vs. asymmetric cryptography, Classical Cryptosystems, Substitution ciphers, Transposition ciphers, Modern Cryptography, Block ciphers (AES, DES),Stream ciphers.	
Module – 2	**08 Hrs**
Public Key Cryptography RSA Algorithm , Key generation, Encryption and decryption, Key Management , Public key infrastructure (PKI), Digital certificates, Elliptic Curve Cryptography (ECC),Basics and applications..	
Module – 3	**08 Hrs**
Cryptographic Hash Functions Hash Functions , Properties and applications, SHA family of hash functions , Message Authentication Codes (MACs) ,HMAC and its applications, Digital Signatures Concepts and algorithms, Verification and applications.	
Module – 4	**08 Hrs**

Network Security Protocols	
Secure Socket Layer (SSL)/Transport Layer Security (TLS), Architecture and operation , Internet Protocol Security (IPsec), Modes of operation, Security associations Virtual Private Networks (VPNs), Concepts and implementations.	
Module – 5	**08 Hrs**
Security Threats and Countermeasures	
Network Security Threats, Types of attacks (DoS, DDoS, phishing), Malware and its impact, Intrusion Detection Systems (IDS), Types and methodologies Firewalls and Security Policies , Types of firewalls, Designing security policies	

Topics for Self learning	
Module – 1	Classical cipher analysis, frequency analysis and basic cryptanalysis
	AES working principles, modes of operation and comparison of block cipher modes
Module – 2	RSA numerical examples, key management and PKI certificate validation
	Elliptic Curve Cryptography and its applications in secure communication
Module – 3	SHA-2/SHA-3 concepts, collision resistance and practical hashing applications
	Digital signature standards and practical signature verification
Module – 4	TLS handshake, certificates and secure session establishment
	IPsec AH/ESP, tunnel and transport modes, and VPN applications
Module – 5	DoS/DDoS attack analysis, phishing and malware case studies
	IDS/IPS, firewall configurations and development of organizational security policies

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the principles of symmetric and asymmetric cryptography and the operation of classical and modern cryptosystems.
CO2	Analyze RSA, ECC, PKI and digital certificates for secure key management and public-key communication.
CO3	Implement and verify cryptographic hash functions, MACs/HMACs and digital signatures for data integrity and authentication.
CO4	Analyze SSL/TLS, IPsec and VPN security protocols and their role in protecting network communications.
CO5	Evaluate network security threats and propose appropriate IDS, firewall and security-policy-based countermeasures.

Textbooks	
1	"Cryptography and Network Security: Principles and Practice", William Stallings, 8th Edition, Pearson Education, 2024, ISBN: 9789357059718.

2	"Cryptography and Network Security", Behrouz A. Forouzan and Debdeep Mukhopadhyay, 3rd Edition, McGraw Hill Education (India) Private Limited, 2015, ISBN: 9789339220945

Reference Books	
1	"Security in Computing", Charles P. Pfleeger, Shari Lawrence Pfleeger and Jonathan Margulies, 5th Edition, Prentice Hall, 2015, ISBN: 9780134085043.
2	"Secrets and Lies: Digital Security in a Networked World", Bruce Schneier, 1st Edition, Wiley, 2000, ISBN: 9780471253112.

Web links and Video Lectures (e – Resources)	
1	https://www.coursera.org/learn/crypto
2	https://www.youtube.com/playlist?list=PLzHjzB7Q0q7WgH3m8pY5MZgRzQm5Z84aC
3	https://www.edx.org/professional-certificate/ritx-cybersecurity-fundamentals
4	https://www.youtube.com/playlist?list=PL5eH0n1q4D9hA6XkzJ5n7F8tT8Gz4r4uG
5	https://www.khanacademy.org/computing/computer-science/cryptography

Semester: II					
HIGH PERFORMANCE COMPUTING					
Course Code	**:**	1MSCS203	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Computer Organization and Architecture, Operating Systems, Data Structures and Algorithms.
Knowledge on Fundamental Concepts	Basic knowledge of computer architecture and parallel processing.
Knowledge on Connected Tools (If any)	OpenMP, MPI, CUDA, GCC/G++

Course Learning Objectives: This course will enable the students to	
CLO1	Infer the fundamental concepts, levels and models of parallelism used in high performance computing.
CLO2	Demonstrate the use of processor architectures, interconnects, communication mechanisms, memory organization, and programming models in HPC systems.
CLO3	Analyze synchronization, scheduling, job allocation, partitioning, dependency and mapping techniques for parallel computation.
CLO4	Asses performance limitations related to bandwidth, latency and power consumption in high performance computing systems.
CLO5	Explore emerging HPC technologies including petascale computing, optical computing, quantum computing and nanotechnology.

Module – 1	**08 Hrs**
Introduction to High Performance Computing: Need for HPC, applications, characteristics of HPC systems, parallel and sequential computing. **Levels of Parallelism**: Instruction-level, transaction-level, task-level, thread-level, memory-level and function-level parallelism. **Models of Parallelism**: SIMD, MIMD, SIMT, SPMD, dataflow models and demand-driven computation. **Parallel Architectures**: N-wide superscalar architectures, multi-core processors and multi-threaded processors.	
Module – 2	**08 Hrs**
Processor Architecture: Organization of HPC processors, superscalar, multicore and multithreaded processors, and accelerated systems.**Interconnect and Communication**: Interconnection mechanisms, communication models, communication latency, bandwidth and overhead.**Memory Organization**: Memory hierarchy, registers, cache, main memory, memory bandwidth, latency, locality and transaction-specific memory design.**Programming Models**: Shared-memory, distributed-memory, message-passing, data-parallel and thread-based programming models. Architectural Case Studies: IBM Cell Broadband Engine, NVIDIA Tesla GPU.**Thread Organization**: Thread creation, organization, execution and mapping.	
Module – 3	**08 Hrs**
Synchronization: Need for synchronization, mutual exclusion, locks, barriers and synchronization overhead. **Scheduling**: Static and dynamic scheduling, load balancing and scheduling overhead. **Job Allocation and Job Partitioning**: Processor allocation, task partitioning, data partitioning and workload distribution. **Dependency Analysis**: Data dependencies, task dependencies, dependency graphs and identification of parallelism. **Mapping Parallel Algorithms onto Parallel Architectures**: Task mapping, data mapping, processor assignment, communication-aware and architecture-aware mapping. **Performance Analysis of Parallel Algorithms**: Execution time, parallel overhead, speedup, efficiency, scalability, Amdahl's Law, Gustafson's Law and performance bottlenecks.	

Module – 4	**08 Hrs**
Bandwidth Limitations: Memory bandwidth, communication bandwidth, bandwidth-bound applications and techniques for reducing bandwidth limitations. **Latency Limitations**: Memory latency, communication latency, sources of latency and effect on parallel applications. **Latency Hiding/Tolerating Techniques**: Multithreading, prefetching, overlapping computation and communication, buffering, and limitations. **Power-aware Processing Techniques**: Power consumption, energy-performance trade-off, power-aware processor design, DVFS and energy-aware scheduling. **Power-aware Memory Design**: Memory power consumption, energy-efficient memory access and low-power memory techniques.	
Module – 5	**08 Hrs**
Petascale Computing: Introduction, petaflop computing, characteristics, challenges, scalability and resource management. **Optics in Parallel Computing**: Optical computing, optical interconnects, optical communication, advantages, limitations and applications. **Quantum Computers**: Qubits, quantum gates, quantum parallelism, classical HPC versus quantum computing and potential applications. **Recent Developments in Nanotechnology**: Nano-scale computing, nanoelectronics, nano-scale memory technologies and impact on HPC	

Topics for Self learning	
Module – 1	GPU-based parallelism and SIMT Dataflow and demand-driven computation.
	Dataflow and demand-driven computation.
Module – 2	Cache coherence and NUMA.
	OpenMP and MPI programming models.
Module – 3	Task graphs and dependency-based parallelism.
	Practical applications of Amdahl's and Gustafson's Laws.
Module – 4	Roofline model.
	Latency-hiding techniques.
Module – 5	Exascale computing and Optical interconnects.
	Quantum computing and nanoscale technologies.

Course Outcomes: After completing the course, the students will be able to	
CO1	Classify different levels and models of parallelism and relate them to contemporary high performance computing architectures.
CO2	Implement suitable processor architectures, interconnects, communication mechanisms, memory organization, and programming models for HPC systems.
CO3	Analyze synchronization, scheduling, job partitioning, dependency and mapping strategies for parallel algorithms.
CO4	Evaluate the effects of bandwidth, latency and power consumption on the performance and energy efficiency of HPC systems.
CO5	Examine emerging HPC technologies including petascale computing, optical computing, quantum computing and nanotechnology.

Text Books	
1	Georg Hager and Gerhard Wellein, “Introduction to High Performance Computing for Scientists and Engineers”, 2nd Edition, CRC Press, 2026, ISBN 978-1482252934.
2	John L. Hennessy and David A. Patterson, “Computer Architecture: A Quantitative Approach”, 6th Edition, Morgan Kaufmann, 2019, ISBN 978-0128119051.
3	Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, “Introduction to Parallel Computing”, 2nd Edition, Pearson, 2003, ISBN 978-0201648652.

Reference Books	
1	David Culler, Jaswinder Pal Singh and Anoop Gupta, “Parallel Computer Architecture: A Hardware/Software Approach”, Morgan Kaufmann, ISBN 978-1558603431.
2	Kai Hwang and Naresh Jotwani, “Advanced Computer Architecture: Parallelism, Scalability, Programmability”, 2nd Edition, McGraw-Hill, ISBN 978-0071133420

3	Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, "Distributed and Cloud Computing: From Parallel Processing to the Internet of Things", Morgan Kaufmann, ISBN 978-0123858801.

Web links and Video Lectures (e – Resources)	
1	NVIDIA CUDA Documentation – https://docs.nvidia.com/cuda/
2	OpenMP API Specification – https://www.openmp.org/specifications/
3	MPI Forum – https://www.mpi-forum.org/
4	High Performance Computing resources – https://www.hpcwire.com/

Semester: II					
Advanced Algorithms					
Course Code	**:**	1MSCS204	**Credits**	**:**	3
Hours/Semester (L:T:P:L)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	40 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Data Structures and Algorithms, Programming in C/C++/Java, Discrete Mathematics
Knowledge on Fundamental Concepts	Basic programming concepts, data structures, algorithm analysis, time and space complexity, recursion, searching and sorting techniques, graphs and trees
Knowledge on Connected Tools (If any)	Familiarity with programming IDEs, compilers, debugging tools, and basic version-control tools such as Git

Course Learning Objectives: This course will enable the students to	
CLO1	Interpret the growth rates of functions and the complexity of algorithms using asymptotic notations and recurrence techniques.
CLO2	Understand linear-time sorting algorithms and advanced algorithm design techniques such as dynamic programming and greedy strategies to solve computational problems.
CLO3	Apply advanced graph algorithms and maximum-flow techniques to solve shortest-path, flow-network, and matching problems.
CLO4	Identify number-theoretic algorithms and modular arithmetic techniques to solve computational and cryptographic problems, including RSA.
CLO5	Analyze and apply advanced data structures and string-matching algorithms to design efficient solutions for complex computational problems.

Module – 1	**08 Hrs**
Analysis techniques: Growth of functions: Asymptotic notation, Standard notations and common functions, Substitution method for solving recurrences, Recursion tree method for solving recurrences, Master theorem. **Amortized Analysis:** Aggregate analysis, the accounting method, the potential method.	
Module – 2	**08 Hrs**
Sorting in Linear Time: Lower bounds for sorting, Counting sort, Radix sort, Bucket sort. **Advanced Design and Analysis Technique:** Matrix-chain multiplication, longest common subsequence, Elements of the greedy strategy, an activity-selection problem	
Module – 3	**08 Hrs**

Graph Algorithms: Bellman-Ford Algorithm, Shortest paths in a DAG, Johnson's Algorithm for sparse graphs. **Maximum Flow:** Flow networks, Ford Fulkerson method and Maximum Bipartite Matching
Module – 4 **08 Hrs**
Number Theoretic Algorithms: Elementary notions, GCD, Modular arithmetic, solving modular linear equations, The Chinese remainder theorem, Powers of an element, RSA cryptosystem
Module – 5 **08 Hrs**
Advanced Data structures: Structure of Fibonacci heaps, Mergeable-heap operations, decreasing a key and deleting a node, Binomial Queues, Splay Trees. **String Matching Algorithms:** Naïve algorithm, Rabin-Karp algorithm, String matching with finite automata, Knuth-Morris-Pratt algorithm.

Topics for Self learning	
Module – 1	Analysis of iterative and recursive algorithms using asymptotic notations
	Applications of amortized analysis in dynamic data structures
Module – 2	Applications of Matrix-Chain Multiplication and Longest Common Subsequence
	Greedy algorithms and their applications in scheduling and optimization problems
Module – 3	Applications of Johnson's Algorithm in sparse graphs
	Maximum Flow and its applications in network routing and bipartite matching
Module – 4	Applications of GCD and the Extended Euclidean Algorithm
	Public-key cryptography and practical applications of the RSA cryptosystem
Module – 5	Applications and performance comparison of Fibonacci Heaps and Binomial Queues
	Splay Trees and self-adjusting data structures

Course Outcomes: After completing the course, the students will be able to	
CO1	Analyze algorithm efficiency using asymptotic notation, recurrence relations, substitution, recursion-tree, and Master theorem methods, along with amortized analysis.
CO2	Apply counting sort, radix sort, bucket sort, dynamic programming, and greedy algorithms to solve optimization and sorting problems efficiently.
CO3	Implement and analyze shortest-path, maximum-flow, and bipartite-matching algorithms for solving graph-based computational problems.
CO4	Apply GCD, modular arithmetic, modular linear equations, Chinese Remainder Theorem, and RSA techniques to solve number-theoretic and cryptographic problems.

CO5	Design and evaluate solutions using Fibonacci heaps, binomial queues, splay trees, and advanced string-matching algorithms such as Rabin-Karp, finite automata, and KMP.

Textbooks	
1	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein; Introduction to Algorithms; Columbia University, 2022,ISBN:978-0262046305
2	Mark Allen Weiss; Data Structures and Algorithm Analysis in C++, Addison-Wesley. 2013,ISBN:978-0132847377

Reference Books	
1	Kozen DC, The design and analysis of algorithms, Springer Science & Business Media. 1992, ISBN:978-0387976877
2	Kenneth A. Berman, Jerome L. Paul,Algorithms, Cengage Learning. 2002

Web links and Video Lectures (e – Resources)	
1	https://nptel.ac.in/courses/106101060
2	https://www.coursera.org/learn/algorithmic-toolbox
3	https://www.coursera.org/learn/algorithms-part1
4	https://www.coursera.org/learn/algorithms-part2
5	https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2011/

Semester: II						
Quantum Computing						
Course Code	**:**	1MSCS205A		**Credits**	**:**	3
Hours/Week (L:T:P)	**:**	42:0:48		**CIE**	**:**	50 Marks
Total Hours	**:**	90		**SEE**	**:**	50 Marks
Course Type	**:**	PEC		**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Basic knowledge of Linear Algebra, Probability and Statistics, Data Structures and Algorithms, Fundamentals of Computer Architecture
Knowledge on Fundamental Concepts	Vectors and matrices, eigenvalues/eigenvectors, Probability theory, Boolean logic and binary computation
Knowledge on Connected Tools (If any)	Python programming

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental principles of quantum mechanics and their role in quantum computation.
CLO2	Explain quantum states, quantum gates, quantum circuits, measurement and quantum entanglement using appropriate mathematical representations.
CLO3	Apply quantum algorithms and circuit-based techniques to solve selected computational problems.
CLO4	Analyze quantum algorithms with respect to their computational complexity and compare them with classical approaches.
CLO5	Explore quantum programming frameworks and develop quantum circuits for implementing basic quantum computing applications.

Module – 1	**08Hrs**
Introduction to Quantum Computing, Evolution from Classical to Quantum Computing, Limitations of Classical Computing, Qubits and Quantum States, Representation of Qubit using Dirac Notation, Bloch Sphere Representation, Superposition Principle, Quantum Measurement, Probability Amplitudes, Quantum States and State Vectors, Tensor Product of Quantum States, Multiple Qubit Systems. **Introduction to Quantum Entanglement**: Bell States, Quantum Correlation and Applications of Entanglement. **Text Book 1**:**Ch-**	
Module – 2	**08Hrs**
Single-Qubit Gates: Pauli-X, Pauli-Y, Pauli-Z, Hadamard Gate, Phase Gates and Rotation Gates. Multi-Qubit Gates: Controlled-NOT, Controlled-Z, SWAP Gate and Toffoli Gate. Quantum Circuits, Circuit Representation, Quantum Measurement Circuits, Reversible Computation, Quantum Circuit Composition, Unitary Operations, Quantum Teleportation, Superdense Coding. **Text Book 1**: **Ch-**	
Module – 3	**08Hrs**
Introduction to Quantum Algorithms, Classical versus Quantum Algorithms. **Deutsch Algorithm**: Problem Definition, Quantum Circuit and Algorithm Analysis. **Deutsch–Jozsa Algorithm**: Problem Definition, Oracle, Quantum Circuit and Computational Ad	

Bernstein–Vazirani Algorithm: Problem Formulation, Quantum Circuit and Algorithm. **Simon's Algorithm**: Problem Definition, Quantum Circuit, Measurement and Period Finding. Introduction to Quantum Fourier Transform and its role in quantum algorithms. **Text Book 1: Ch**
Module – 4 **08Hrs**
Advanced Quantum Algorithms **Grover's Search Algorithm**: Search Problem, Oracle Construction, Amplitude Amplification, Quantum Circuit and Complexity Analysis. **Quantum Fourier Transform**: Definition, Circuit Construction and Applications. **Quantum Phase Estimation**: Principle, Quantum Circuit and Applications. **Shor's Algorithm**: Integer Factorization Problem, Period Finding, Quantum Fourier Transform and Computational Complexity. Comparison of Classical and Quantum Computational Complexity. **Text Book 1: Ch**
Module – 5 **10Hrs**
Quantum Programming and Emerging Applications Quantum Programming Concepts, Quantum Software Stack, Quantum Circuit Programming. Introduction to Qiskit, Quantum Registers and Classical Registers, Quantum Circuit Construction, Applying Quantum Gates, Measurement and Circuit Execution, Quantum Simulators and Real Quantum Hardware.Implementation of Basic Quantum Algorithms using Qiskit. Applications of Quantum Computing: Cryptography, Optimization, Machine Learning, Drug Discovery, Financial Computing and Scientific Computing. Introduction to Quantum Error and Noise, Decoherence, Quantum Error Correction and Challenges in Practical Quantum Computing. **Text Book 1**: **Ch-**

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the fundamental principles of quantum computation, qubits, quantum states, superposition and entanglement.
CO2	Illustrate quantum gates and construct quantum circuits for implementing quantum operations and communication protocols.
CO3	Apply fundamental quantum algorithms such as Deutsch–Jozsa, Bernstein–Vazirani and Simon's algorithm to solve computational problems.
CO4	Analyze Grover's, Quantum Fourier Transform, Phase Estimation and Shor's algorithms with respect to their computational complexity and quantum advantage.
CO5	Develop and execute quantum circuits using quantum programming frameworks and evaluate their applications and practical challenges.

Text Books	
1	Quantum Computation and Quantum Information, Michael A. Nielsen and Isaac L. Chuang, 10th Anniversary Edition, Cambridge University Press, 2010.

Reference Books	
1	Quantum Computing for Everyone, Chris Bernhardt, MIT Press, 2019.
2	Quantum Computing: An Applied Approach, Jack D. Hidary, Springer, 2019.

Web links and Video Lectures (e – Resources)	
1	https://quantum.cloud.ibm.com/learning/en?utm_source=chatgpt.com
2	https://quantum.cloud.ibm.com/learning/en/modules?utm_source=chatgpt.com
3	https://quantum.cloud.ibm.com/learning/en/courses?utm_source=chatgpt.com

Semester: II						
Fog & Edge Computing						
Course Code	**:**	**1MSCS205B**		**Credits**	**:**	3
Hours/Week(L:T:P)	**:**	42:0:48		**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs		**SEE**	**:**	50 Marks
Course Type	**:**	PEC		**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Computer Networks, Cloud Computing, Internet of Things, Basics of Machine Learning
Knowledge on Fundamental Concepts	**Computer Networks:** Basic networking concepts, TCP/IP, IP addressing, routing, latency, and bandwidth. **Cloud Computing:** Cloud models, virtualization, resource sharing, storage, and basic cloud services. **Internet of Things (IoT):** IoT architecture, sensors, actuators, gateways, and basic communication protocols.
Knowledge on Connected Tools	NS2 simulators, iFogSim, EdgeCloudSim

Course Learning Objectives: This course will enable the students to	
CLO1	**Explain** the fundamental concepts, architectures, characteristics, and applications of **Fog and Edge Computing** in distributed IoT environments.
CLO2	**Analyze and design** Fog/Edge computing solutions integrated with **5G/6G networks** for low-latency, scalable, and real-time applications.
CLO3	**Apply** containerization, middleware, and **AI/ML techniques at the Edge** to develop efficient and intelligent IoT applications.
CLO4	**Evaluate and implement** secure and resource-efficient Fog/Edge solutions considering **latency, energy consumption, resource utilization, QoS, and security** for real-time applications.

Module – 1	**08 Hrs**

Foundations of Fog and Edge Computing

Introduction to Fog Computing – Relevant Technologies, Fog and Edge Computing Completing the Cloud, Hierarchy of Fog and Edge Computing, Business Models, Opportunities and Challenges. Addressing the Challenges in Federating Edge Resources – The Networking Challenge, The Management Challenge, Miscellaneous Challenges. The Edge-Cloud Continuum and Emerging Paradigms – Cloud-Fog-Edge-Device continuum, on-device/near-device processing, and the shift toward decentralized, privacy-preserving computation. Trending Directions – market and technology outlook for fog/edge (rising adoption of open-source edge stacks), sustainable and energy-efficient (green) fog/edge deployments, and the roadmap toward 6G-enabled fog/edge computing as highlighted by recent IEEE ComSoc and ICFEC/FMEC research agendas 8 hours

Module – 2	08 Hrs

Integrating IoT, Fog, and Cloud Infrastructures

Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds – 5G Fundamentals, Network Slicing in 5G, Network Slicing in Software-Defined Clouds, Network Slicing Management in Edge and Fog. Trending Directions – convergence of terrestrial and non-terrestrial (satellite) networks with edge infrastructure, early network-slicing and orchestration concepts for 6G fog/edge, and Open RAN-based edge deployments supporting ultra-low-latency, high-reliability use cases such as AR/VR, autonomous driving, and Industry 4.0 8 hours

Module – 3	08 Hrs

Optimization and Middleware for Fog and Edge Computing

Optimization Problems in Fog and Edge Computing – Preliminaries, The Case for Optimization in Fog Computing, Optimization Opportunities along the Fog Architecture, Optimization Opportunities along the Service Life Cycle. Middleware for Fog and Edge Computing: Design Issues – Need for Fog and Edge Computing Middleware, Design Goals, State-of-the-Art Middleware Infrastructures, System Model, Architecture Examples, Future Research Directions. Trending Directions – serverless (Function-as-a-Service) computing at the edge, edge-native orchestration platforms (KubeEdge, K3s, OpenYurt) for managing distributed lightweight clusters, and the emerging concept of edge digital twins for real-time monitoring and optimization of physical assets.

Module – 4	08 Hrs

Lightweight Container Middleware and Data Management Data Management in Fog Computing – Fog Data Management, Predictive Analysis to Support Fog Application Deployment, Predictive Analysis with Fortich. Trending Directions – container-based edge clusters using Docker/Kubernetes-derivative runtimes for resource-constrained nodes, data governance and caching strategies for high-velocity IoT data streams, and zero-trust security architectures adapted from enterprise environments for smart-home and industrial edge deployments

Module – 5	08 Hrs

Artificial Intelligence and Machine Learning at the Edge

Using Machine Learning for Protecting the Security and Privacy of Internet of Things (IoT) Systems – Introduction, Survey of ML Techniques for Defending IoT Devices, Machine Learning in Fog Computing. Fog AI – shifting computational intelligence from centralized clouds to fog nodes equipped with neural processing units for real-time, low-power inference. Trending Directions – TinyML and ultra-compact/quantized models (including tiny language models) for microcontrollers and wearables, federated and on-device fine-tuning for privacy-preserving personalization, dedicated edge AI accelerators and neuromorphic chips (e.g., Loihi-class, NPUs), and forward-looking research frontiers such as quantum computing at the edge and edge/fog computing in space-based (satellite) networks

Course Outcomes

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

Course Outcomes: After completing the course, the students will be able to	
CO1	Understand the foundational concepts, architectures, and challenges of fog and edge computing, and their positioning in the emerging edge-cloud continuum.

CO2	Analyze IoT-Fog-Cloud integration models and evaluate resource management and network-slicing strategies across 5G/6G and software-defined infrastructures.
CO3	Demonstrate the use of middleware, orchestration frameworks, and optimization techniques for service placement, data management, and application deployment across the edge-fog-cloud continuum.
CO4	Design and deploy lightweight, containerized, and serverless architectures for edge-cloud systems, with attention to storage, orchestration, and security management.
CO5	Apply machine learning, federated learning, and TinyML techniques to build intelligent, secure, and privacy-preserving fog/edge applications, and appraise emerging directions such as edge digital twins and AI accelerators.

Topics for Self learning	
Module – 1	Edge–Cloud Continuum and decentralized computing architectures.
	Green Edge Computing, privacy-preserving edge computing, and 6G-enabled Fog/Edge systems.
Module – 2	5G/6G Network Slicing and orchestration in Fog and Edge environments.
	Open RAN, satellite–edge integration, and ultra-low-latency applications
Module – 3	Serverless Edge Computing and Function-as-a-Service (FaaS).
	KubeEdge, K3s, OpenYurt, and Edge Digital Twins
Module – 4	Docker/Kubernetes-based lightweight edge clusters and container orchestration.
	Edge data caching, data governance, and Zero-Trust security for IoT applications.
Module – 5	TinyML, quantized models, and on-device AI for resource-constrained devices.
	Federated learning, edge AI accelerators, neuromorphic computing, and AI at the edge.

Learning Resources

Textbooks	
1	1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley Publications, 2019.
2	Simone Cirani, Gianluigi Ferrari, Marco Picone, Luca Veltri, "Internet of Things – Architectures, Protocols and Standards", John Wiley & Sons Ltd., 2019

Reference Books	
1	Wei Chang, Jie Wu (Eds.), "Fog/Edge Computing for Security, Privacy and Applications", Springer, 2021.
2	Pethuru Raj, Anupama Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press.
3	1. Recent proceedings of IEEE ICFEC (International Conference on Fog and Edge Computing) and FMEC (International Conference on Fog and Mobile Edge Computing) for current research trends in 6G-enabled edge, edge digital twins, and sustainable fog/edge computing.

	2. Recent survey/field-guide literature on TinyML, federated learning, and on-device generative AI (e.g., TensorFlow Lite Micro, Flower, TensorFlow Federated frameworks) for current developments in edge AI.

Web links and Video Lectures (e – Resources)	
1	https://www.nptel.ac.in/courses/106104449?utm_source=chatgpt.com,
2	https://onlinecourses.nptel.ac.in/noc25_cs75/preview?utm_source=chatgpt.com
3	https://nptel.ac.in/courses/106105166?utm_source=chatgpt.com

Semester: II					
GENETIC ALGORITHMS					
Course Code	**:**	1MSCS205C	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Data Structures and Algorithms, Artificial Intelligence, Machine Learning
Knowledge on Fundamental Concepts	Basic programming concepts, optimization techniques, probability, statistics, search algorithms, and mathematical functions
Knowledge on Connected Tools (If any)	Python/MATLAB, simulation tools, and basic knowledge of optimization or machine-learning libraries such as NumPy and SciPy

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental concepts, principles, and terminology of Genetic Algorithms (GAs).
CLO2	Apply genetic operators such as selection, crossover, and mutation to solve optimization problems.
CLO3	Analyze different Genetic Algorithm representations, fitness functions, and selection strategies
CLO4	Implement Genetic Algorithms using suitable programming tools for real-world optimization problems.
CLO5	Evaluate the performance of Genetic Algorithms and apply advanced GA techniques to engineering and computational problems.

Module – 1	**08 Hrs**
Fundamentals of Genetic Algorithms : Introduction to Evolutionary Computation: Evolutionary computation, biological evolution, natural selection and survival of the fittest. Genetic Algorithms: Definition, characteristics, terminology, advantages and limitations. Genetic Algorithm Components: Population, chromosome, gene, allele, genotype, phenotype and fitness function. Representation Techniques: Binary, integer, real-valued and permutation representations. Genetic Algorithm Workflow: Initialization, fitness evaluation, selection, genetic operations and termination criteria. Applications of Genetic Algorithms.	
Module – 2	**08 Hrs**
Genetic Algorithm Operators : Fitness Evaluation: Objective function, fitness function, fitness scaling and normalization. Selection Techniques: Roulette wheel selection, rank selection, tournament selection and elitism. Crossover Operators: Single-point, two-point, multi-point and uniform crossover. Mutation Operators: Bit-flip, swap, inversion and Gaussian mutation. Replacement Strategies: Generational replacement, steady-state replacement and elitism. Genetic Algorithm parameter settings: Population size, crossover probability, mutation probability and number of generations.	
Module – 3	**08 Hrs**
Genetic Algorithm Design and Implementation :Genetic Algorithm Framework: Algorithm structure, flowchart and pseudocode. Population Initialization: Random and heuristic initialization. Constraint Handling: Penalty functions, repair methods and feasibility-based	

approaches. Convergence: Termination criteria, premature convergence, diversity preservation and exploration–exploitation trade-off. Parameter Tuning: Selection pressure, crossover rate, mutation rate and population size. Implementation of Genetic Algorithms using Python/MATLAB. Analysis of Genetic Algorithm performance.

Module – 4	**08 Hrs**

Genetic Algorithms for Optimization : Optimization Problems: Single-objective, multi-variable, continuous and discrete optimization. Combinatorial Optimization: Travelling Salesman Problem, Knapsack Problem and scheduling problems. Function Optimization: Mathematical function optimization using Genetic Algorithms. Constraint Optimization: Handling equality and inequality constraints. Applications: Resource allocation, engineering design, network optimization and scheduling. Performance Analysis: Fitness convergence, solution quality, computational cost and comparison with conventional optimization techniques.

Module – 5	**08 Hrs**

Advanced Genetic Algorithms and Applications : Multi-objective Genetic Algorithms: Multiple objectives, Pareto optimality, Pareto dominance and Pareto front. Hybrid Genetic Algorithms: Integration of Genetic Algorithms with local search and other optimization techniques. Genetic Programming: Basic concepts, representation and applications. Genetic Algorithms with Machine Learning: Feature selection, hyperparameter optimization and model optimization. Applications of Genetic Algorithms: Cloud computing, data mining, engineering, scheduling and network optimization. Recent trends and research directions in Genetic Algorithms.

Topics for Self learning	
Module – 1	Evolutionary computation, genetic representation and real-world applications of Genetic Algorithms.
Module – 2	Advanced selection methods, crossover and mutation strategies.
Module – 3	Genetic Algorithm implementation using Python and parameter optimization.
Module – 4	Genetic Algorithm applications to TSP, Knapsack and scheduling problems.
Module – 5	Multi-objective optimization, hybrid Genetic Algorithms and Genetic Programming.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the fundamental concepts, terminology, components, and working principles of Genetic Algorithms.
CO2	Apply appropriate selection, crossover, mutation, and replacement techniques to solve optimization problems.
CO3	Analyze Genetic Algorithm parameters, convergence behavior, constraints, and solution quality.
CO4	Implement and evaluate Genetic Algorithm-based solutions using suitable programming tools.
CO5	Examine advanced Genetic Algorithm techniques and their applications in real-world optimization problems.

Text Books

1	David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989, ISBN 978-0201157673.
2	Melanie Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1998, ISBN 978-0262631853.
3	Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer, 1996, ISBN 978-3540606765.

Reference Books	
1	Kenneth A. De Jong, "Evolutionary Computation: A Unified Approach", MIT Press, 2006, ISBN 978-0262041944.
2	David E. Goldberg, "The Design of Innovation: Lessons from and for Competent Genetic Algorithms", Springer, 2002, ISBN 978-1402070980.
3	Thomas Bäck, "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms", Oxford University Press, 1996, ISBN 978-0195099713.

Web links and Video Lectures (e – Resources)	
1	https://www.geeksforgeeks.org/dsa/genetic-algorithms/
2	https://www.mathworks.com/discovery/genetic-algorithm.html?utm_source=chatgpt.com
3	https://nptel.ac.in/?utm_source=chatgpt.com
4	https://www.youtube.com/results?search_query=genetic+algorithm+lectures

Semester: II						
Secure Coding						
Course Code	**:**	1MSCS205D		**Credits**	**:**	3
Hours/Week (L:T:P)	**:**	3:0:0		**CIE**	**:**	50 Marks
Total Hours	**:**	42+48		**SEE**	**:**	50 Marks
Course Type	**:**	PEC		**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Fundamentals of Programming, Object-Oriented Programming, Database Management Systems, Computer Networks
Knowledge on Fundamental Concepts	Programming constructs, data structures, databases, operating-system basics, networking fundamentals and software development practices
Knowledge on Connected Tools (If any)	C/C++ or Python/Java programming, Git, browser developer tools

Course Learning Objectives: This course will enable the students to	
CLO1	Understand secure coding principles, software security requirements, common vulnerabilities and secure software development practices.
CLO2	Explain input validation, authentication, authorization, session management, cryptographic protection and secure data-handling techniques.
CLO3	Apply secure programming practices to prevent common application vulnerabilities including injection, cross-site scripting, broken access control and insecure configurations.
CLO4	Analyze source code and software components to identify security weaknesses, attack surfaces and potential exploitation paths.
CLO5	Explore secure software development life cycle, security testing, code review, vulnerability management and DevSecOps practices.

Module – 1	**08Hrs**
Introduction to Secure Coding, Software Security and Secure Software Development, Security Principles: Confidentiality, Integrity, Availability, Authentication, Authorization and Accountability. Secure Coding Principles: Least Privilege, Defense in Depth, Fail-Safe Defaults, Complete Mediation, Separation of Duties, Secure Defaults and Minimize Attack Surface. Threat Modeling and Risk Assessment: Assets, Threats, Vulnerabilities, Attack Surface, Trust Boundaries, STRIDE overview and Security Requirements. Common Coding Vulnerabilities: Improper Input Validation, Injection, Buffer Overflow, Race Conditions, Hard-Coded Credentials, Insecure Configuration and Improper Error Handling. Introduction to OWASP Top 10 and CWE; secure coding standards and security-aware programming practices. Text Book 1: Ch1,2 3	
Module – 2	**08Hrs**

Input Validation and Data Sanitization: Whitelisting, Blacklisting, Canonicalization, Output Encoding and Safe Data Handling.
Injection Vulnerabilities: SQL Injection, Command Injection, LDAP Injection and Prevention Techniques.
Authentication and Credential Security: Password Storage, Salting and Hashing, Multi-Factor Authentication, Account Lockout and Credential Management.
Authorization and Access Control: Role-Based Access Control, Principle of Least Privilege, Access-Control Checks and Insecure Direct Object References.
Session Security: Session IDs, Cookies, Session Fixation, Session Hijacking, Timeout and Secure Session Management.
Secure Error Handling, Logging and Auditing.
Text Book 1: Ch 4,5,6

Module – 3	**08Hrs**

Secure Web Application Coding: HTTP/HTTPS, Security Headers, Same-Origin Policy and Secure Browser Communication. Cross-Site Scripting (XSS): Reflected, Stored and DOM-based XSS; Prevention using Output Encoding and Content Security Policy.
Cross-Site Request Forgery (CSRF): Attack Mechanism and Prevention using Tokens and SameSite Cookies. File and Resource Security: Secure File Upload, Path Traversal, File Inclusion, Temporary Files and Resource Permissions.
Database Security in Application Code: Parameterized Queries, Prepared Statements, Secure Database Connectivity and Protection of Sensitive Data.
Secure API Development: API Authentication, Authorization, Input Validation, Rate Limiting and Secure API Error Responses.
Text Book 1: Ch 10, 13

Module – 4	**09Hrs**

Memory and Program Security: Buffer Overflow, Stack/Heap Issues, Integer Overflow, Format String Vulnerabilities and Memory-Safe Programming Practices.
Cryptographic Coding Practices: Encryption versus Hashing, Symmetric and Asymmetric Cryptography, Digital Signatures, Key Management, Secure Randomness and Avoiding Weak Cryptographic Algorithms.
Data Protection: Data at Rest and in Transit, Secure Storage, Secrets Management, Token Protection and Privacy-Aware Coding.
Concurrency and System Security: Race Conditions, Time-of-Check/Time-of-Use, Deadlocks and Secure Resource Access.
Secure Dependency and Configuration Management: Third-Party Libraries, Dependency Vulnerabilities, Patch Management, Secure Configuration and Secrets in Source Code.
Secure Coding in C/C++ and Managed Languages: Language-specific security considerations and compiler/runtime protections.
Text Book 1: Ch 13, 14

Module – 5	10Hrs

Secure Software Development Life Cycle (SSDLC): Security Planning, Requirements, Secure Design, Implementation, Verification, Deployment and Maintenance.
Secure Code Review: Review Process, Security Checklists, Manual Review Techniques and Defect Classification.
Security Testing: Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA), Fuzz Testing and Vulnerability Scanning.
DevSecOps: Security in CI/CD Pipelines, Automated Security Checks, Dependency Scanning, Container and Configuration Security.
Vulnerability Management: CVE, CWE, CVSS, Vulnerability Disclosure, Remediation, Regression Testing and Security Metrics.
Case Studies and Hands-on Exercises: Identification and remediation of vulnerable code using representative applications and security testing tools.
Text Book 1: Ch 15

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain secure coding principles, software security requirements, common vulnerabilities and secure software development practices.
CO2	Apply secure programming techniques for input validation, authentication, authorization, session management and secure data handling.
CO3	Analyze source code and applications to identify common security vulnerabilities, attack surfaces and potential exploitation paths.
CO4	Develop secure software components using appropriate cryptographic practices, secure APIs, memory-safe techniques and secure configuration.
CO5	Evaluate software security using code review, security testing, vulnerability assessment and DevSecOps practices.

Text Books	
1	Secure Coding in C and C++, Robert C. Seacord, 2nd Edition, Addison-Wesley Professional, 2013.

Reference Books	
1	Secure Coding: Principles and Practices, Mark G. Graff and Kenneth R. van Wyk, O'Reilly Media, 2003.
2	Secure by Design, Daniel Deogun, Dan Bergh Johnsson and Daniel Sawano, Manning Publications, 2019.

Web links and Video Lectures (e – Resources)	
1	https://owasp.org/www-project-top-ten/
2	https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/
3	https://cwe.mitre.org/

Semester: II					
Software Defined Networks(SDN)					
Course Code	:	1MSCS206A	**Credits**	:	3
Hours/Semester (L:T:P:SL)	:	42:0:28:50	**CIE**	:	50Marks
Total Hours	:	120	**SEE**	:	50Marks
Course Type	:	Elective	**SEE Duration**	:	3hr

Prerequisites:	
Courses	Computer Networks, Operating Systems, Programming Fundamentals
Knowledge on Fundamental Concepts	TCP/IP and OSI models, IP addressing, routing and switching, Ethernet/VLANs, packet forwarding, control and data planes, basic Linux, and programming fundamentals.
Knowledge on Connected Tools (If any)	Network simulation/emulation tools such as Cisco Packet Tracer, Mininet,

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamentals of packet switching, data-center networking, and the need for Software Defined Networking (SDN).
CLO2	Explain the evolution, architecture, principles, and interoperability requirements of SDN.
CLO3	Analyze SDN operation, including controllers, devices, applications, and OpenFlow protocol features.
CLO4	Compare alternative SDN approaches such as API-based SDN, overlay networks, and hypervisor-based virtualization.
CLO5	Evaluate SDN deployment models and use cases for modern data-center networks.

Module – 1	**08 Hrs**
Introduction, Basic Packet Switching Terminology, The Modern Data Center, Traditional Switch Architecture, Autonomous and Dynamic Forwarding Tables. **Why SDN?** Evolution of Switches and Control Planes, Cost, SDN Implications for Research and Innovation, Data Center Innovation, Data Center Needs.	
Module – 2	**08 Hrs**
Genesis of SDN The Evolution of Networking Technology, Forerunners of SDN, Legacy Mechanisms Evolve Toward SDN, Software Defined Networking is Born, Sustaining SDN Interoperability, Open Source Contributions, Network Virtualization, May I Please Call My Network SDN?	
Module – 3	**08 Hrs**
How SDN Works Fundamental Characteristics of SDN, SDN Operation, SDN Devices, SDN Controller, SDN Applications, Alternate SDN Methods. **The OpenFlow Specification** Specific Terminology, OpenFlow Overview, OpenFlow 1.0 and OpenFlow Basics, OpenFlow 1.1 Additions, OpenFlow 1.2 Additions, OpenFlow 1.3 Additions, OpenFlow 1.4 Additions, OpenFlow 1.5 Additions, Improving OpenFlow Interoperability, Optical Transport Protocol Extensions, OpenFlow Limitations.	
Module – 4	**08 Hrs**
Alternative Definitions of SDN Potential Drawbacks of Open SDN, SDN via API, SDN via Hypervisor-Based Overlays.	

Emerging Protocol, Controller, and Application Models, Expanded Definitions of SDN, Additional SDN Protocol Models, Additional SDN Controller Model, Additional Application Models, New Approaches to SDN Security.	
Module – 5	**08 Hrs**
SDN in the Data Center: Data Center Definition, Data Center Demands, Tunnelling Technologies for the Data Center, Path Technologies in the Data Center, Ethernet Fabrics in the Data Center, SDN Use Cases in the Data Center, Comparison of Open SDN, Overlays, and APIs.	

Course Outcomes: After completing the course, the students will be able to	
CO1	Interpret packet-switching concepts, traditional switch architecture, control planes, and the motivation for adopting SDN in data centers.
CO2	Describe the evolution of networking technologies, forerunners of SDN, network virtualization, open-source contributions, and SDN interoperability.
CO3	Analyze the architecture and operation of SDN, including SDN devices, controllers, applications, alternate SDN methods, and OpenFlow specifications.
CO4	Compare Open SDN with alternative SDN definitions and models, including API-based SDN, hypervisor-based overlays, emerging controllers, applications, and security approaches.
CO5	Evaluate SDN applications in data centers, including tunnelling, path technologies, Ethernet fabrics, and the comparative suitability of Open SDN, overlays, and APIs.

Text Books	
1	Software Defined Networks, A Comprehensive Approach Second Edition, 2017, Paul Goransson Chuck Black, Timothy Culver, ISBN: 978-0-12-804555-8.

Reference Books	
1	Foundations of Modern Networking SDN, NFV, QoE, IoT, and Cloud William Stallings, 2016,by Pearson Education, ISBN-13: 978-0-13-417539-3 , ISBN-10: 0-13-417539-5
2	Thomas D Nadeau, Ken Gray, “SDN: Software Defined Networks”, O’Reilly Media, 2013.
3	Oswald Coker, Siamak Azodolmolky, “Software-Defined Networking with OpenFlow”, 2nd Edition, O’Reilly Media, 2017.

Web links and Video Lectures (e – Resources)	
1	https://www.youtube.com/watch?v=nUN40N91fJ4&list=PLJ6vZHpRUoJNn-3F918wKpAtr6gGUOixc
2	https://www.youtube.com/watch?v=CaukSKg_sI0

Semester: II					
Advanced Tentive Models					
Course Code	**:**	1MSCS206B	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Machine Learning, Deep Learning, Probability and Statistics, Linear Algebra and Python Programming.
Knowledge on Fundamental Concepts	neural networks, optimization, representation learning, probability distributions, and Python-based machine learning frameworks.
Knowledge on Connected Tools (If any)	Python, Jupyter Notebook/Google Colab, PyTorch or TensorFlow

Course Learning Objectives: This course will enable the students to	
CLO1	Understand the fundamental concepts, architectures and learning principles of advanced generative and representation models.
CLO2	Analyze deep generative models and their training procedures, including autoencoders, variational models and adversarial learning.
CLO3	Apply advanced generative modelling techniques for image, text, audio and multimodal data generation.
CLO4	Evaluate transformer-based and diffusion-based generative models with respect to quality, efficiency, controllability and robustness.
CLO5	Develop and assess practical applications of advanced generative models using suitable datasets, frameworks and evaluation metrics.

Module – 1	**08 Hrs**
Foundations of Advanced Generative Models Introduction to generative modelling, discriminative vs. generative learning, probability distributions and likelihood-based learning, latent-variable models, representation learning, autoencoders, denoising autoencoders, sparse and contractive autoencoders, variational autoencoders (VAE), evidence lower bound (ELBO), latent-space representations and applications	
Module – 2	**08 Hrs**
Generative Adversarial Models Generative Adversarial Networks (GANs), generator and discriminator architecture, minimax objective, training procedure, DCGAN, conditional GAN, Wasserstein GAN, WGAN-GP, CycleGAN, StyleGAN concepts, mode collapse, instability, evaluation of GAN-generated samples and practical applications	
Module – 3	**08 Hrs**

Transformer-Based Generative Models

Sequence modelling and attention, self-attention and multi-head attention, positional encoding, transformer architecture, encoder-decoder transformers, autoregressive language modelling, pre-training and fine-tuning, prompt-based generation, large language model concepts, text generation, summarization and code generation, limitations and responsible use

Module – 4	08 Hrs

Diffusion and Score-Based Generative Models

Introduction to diffusion models, forward and reverse diffusion processes, noise schedules, denoising objectives, DDPM, DDIM, score-based generative modelling, conditional diffusion, classifier and classifier-free guidance, latent diffusion concepts, text-to-image generation and applications in computer vision.

Module – 5	08 Hrs

Multimodal Models, Evaluation and Applications

Multimodal generative learning, vision-language models, image captioning and visual question answering, text-to-image and image-to-text generation, retrieval-augmented generation concepts, model evaluation metrics, FID, IS, BLEU and ROUGE, hallucination and bias, privacy and security issues, deployment considerations and real-world applications

Topics for Self learning	
Module – 1	ELBO derivation, latent-space interpolation, VAE implementation and comparison of autoencoder variants.
Module – 2	GAN training diag nostics, WGAN/WGAN-GP, conditional generation, image-to-image translation and StyleGAN Concepts
Module – 3	Transformer implementation, tokenization, attention visualization, prompt engineering and parameter-efficient fine-tuning concepts.
Module – 4	Diffusion sampling, DDIM, guidance techniques, latent diffusion and practical text-to-image generation.
Module – 5	ultimodal generation, RAG concepts, generative-AI evaluation, safety, bias, hallucination and deployment case studies.

Course Outcomes: After completing the course, the students will be able to	
CO1	Explain the principles and architectures of advanced generative models and latent-variable learning.
CO2	Analyze and implement GAN, VAE and transformer-based generative models for appropriate data-generation tasks.
CO3	Apply diffusion and score-based models for controlled generation of images and other data modalities.
CO4	Evaluate generative models using quantitative and qualitative metrics and analyze their limitations.

CO5	Design and assess generative-AI solutions for multimodal and real-world applications considering ethical, security and deployment issues.

Textbooks	
1	Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press, 2016.
2	David Foster, "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play", 2nd Edition, O'Reilly Media, 2023.
3	Jay Alammar and Maarten Grootendorst, "Hands-On Large Language Models", O'Reilly Media, 2024.

Reference Books	
1	Christopher M. Bishop and Hugh Bishop, "Deep Learning: Foundations and Concepts", Springer, 2024.
2	Simon J. D. Prince, "Understanding Deep Learning", MIT Press, 2023.
3	Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 3rd Edition, O'Reilly Media, 2022.

Web links and Video Lectures (e – Resources)	
1	https://www.deeplearningbook.org/
2	https://pytorch.org/tutorials/
3	https://huggingface.co/learn
4	https://www.tensorflow.org/tutorials
5	https://www.deeplearning.ai/

Semester: II					
Prompt Engineering					
Course Code	**:**	1MSCS206C	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90 Hrs.	**SEE**	**:**	50 Marks
Course Type	**:**	PEC	**SEE Duration**	**:**	03 Hours

Prerequisites:	
Courses	Fundamentals of Artificial Intelligence and Machine Learning, Natural Language Processing basics, Python programming.
Knowledge on Fundamental Concepts	Basic concepts of large language models, tokens, inference, and natural language processing.
Knowledge on Connected Tools (If any)	Jupiter NoteBook

Course Learning Objectives: This course will enable the students to	
CLO1	Describe the principles and concepts underlying prompt engineering.
CLO2	Design and formulate effective prompts for various AI models to achieve desired outputs
CLO3	Analyse and assess the performance of different prompts to improve the quality and reliability of AI-generated outputs.
CLO4	Apply prompt engineering techniques to solve real-world problems in various domains.
CLO5	Collaborate to develop, document, and present prompt-based solutions while considering model safety, responsible AI, and emerging prompting approaches.

Module – 1	**08 Hrs**
Introduction to Prompt Engineering Raise of Context Learning, Prompts, Prompt Engineering, LLM Settings, Basics of prompting, Elements of a Prompt, Settings for Prompting Language Model , General Tips for Designing Prompts, Designing Prompts for Different Tasks: few examples of common tasks using different prompts- Text Summarization, Information Extraction, Question Answering, Text Classification, Conversation/Role Playing, Code Generation, Reasoning.	
Module – 2	**08 Hrs**
Techniques for Effective Prompts Techniques designed to improve performance on complex tasks - Zero-Shot Prompting, Few-shot prompting, Chain-of-thought (CoT) prompting, Zero-Shot CoT, Self-Consistency, Knowledge Generation Prompting, Program-aided Language Model (PAL), ReAct, Directional Stimulus Prompting.	
Module – 3	**08 Hrs**
Best Practices in Prompt Engineering Tools & IDEs Capabilities include: Developing and experimenting with prompts, Evaluating prompts. Versioning and deploying prompts; Advanced prompting techniques: advanced applications	

with LLMs LLMs and external tools/APIs -- LLMs with External Tools; Data-augmented Generation – Steps, External Data, QA with sources, Summarization using sources.
Module – 4 **08 Hrs**
Applications of Prompt Engineering: LLM Applications: Function Calling with LLMs - Getting Started with Function Calling, Function Calling with GPT-4, Function Calling with Open-Source LLMs, Function Calling Use Cases: Conversational Agents, Natural Language Understanding, Math Problem Solving, API Integration, Information Extraction.
Module – 5 **08 Hrs**
Opportunities and Future Directions Model safety, Prompt Injection, Prompt Leaking, Jail Breaking; Reinforcement Learning from Human Feedback (RLHF) -- Popular examples: aClaude (Anthropic), ChatGPT (OpenAI), **Future directions:** Augmented LMs, Emergent ability of LMs, Acting / Planning - Reinforcement Learning, Multimodal Prompting, Graph Prompting

Topics for Self learning	
Module – 1	Compare tokenization and context windows across LLMs.
	Develop prompts for text classification and information extraction.
Module – 2	Compare zero-shot, few-shot, and self-consistency prompting on a common task.
	Document results.
Module – 3	Explore prompt version control and evaluation rubrics.
	Build a source-grounded question-answering workflow.
Module – 4	Develop a small function-calling assistant using a public API or simulated tool
	Test error handling.
Module – 5	Study prompt-injection defenses and responsible AI.
	Explore multimodal and graph prompting.

Course Outcomes: After completing the course, the students will be able to	
CO1	Demonstrate an understanding of prompt engineering principles including how prompt structure and phrasing impact the performance of AI models.
CO2	Design and implement effective prompts- to create and apply prompts for various natural language processing (NLP) tasks, such as text generation, summarization, and translation, using AI models.
CO3	Critically evaluate the effectiveness of prompts - assess the quality and performance of prompts in terms of accuracy, coherence, and relevance, identifying areas for improvement.
CO4	Apply prompt engineering techniques in real-world scenarios - use prompt engineering strategies to address practical problems in domains such as education, healthcare, and business, demonstrating the applicability of AI-driven solutions.
CO5	Collaborate on projects involving prompt engineering - work effectively in teams to design, implement, and evaluate prompt-based solutions, showcasing their ability to contribute to complex AI-related projects.

Textbooks	
1	"Unlocking the Secrets of Prompt Engineering: Master the art of creative language generation to accelerate your journey from novice to pro", Gilbert Mizrahi, Jan 2024, 1st Edition, Packt Publishing, ISBN-13:978-1835083833
2	"Prompt Engineering for Generative AI", James Phoenix, Mike Taylor, May 2024, O'Reilly Media, Inc.,ISBN: 9781098153434

3	"Prompt Engineering for LLMs", John Berryman, Albert Ziegler, O'Reilly Media, Inc. Dec 2024, ISBN: 9781098156152

Reference Books	
1	"The Art of Asking ChatGPT for High-Quality Answers: A Complete Guide to Prompt Engineering", Ibrahim John, Nzunda Technologies Limited, 2023, ISBN-13: 9781234567890.
2	"Programming Large Language Models with Azure Open AI: Conversational programming and prompt engineering with LLMs", Francesco Esposito, Microsoft Pr, 1st Edition, April 2024, ISBN-13: 978-0138280376.

Web links and Video Lectures (e – Resources)	
1	DAIR.AI Prompt Engineering Guide: https://www.promptingguide.ai/
2	OpenAI Prompt Engineering: https://platform.openai.com/docs/guides/prompt-engineering
3	Anthropic Prompt Engineering Overview: https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview

Semester: II					
AI IN DIGITAL TWINNING					
Course Code	**:**	1MSCS206D	**Credits**	**:**	3
Hours/Semester (L:T:P:SL)	**:**	42:0:0:48	**CIE**	**:**	50 Marks
Total Hours	**:**	90	**SEE**	**:**	50 Marks
Course Type	**:**	PCC	**SEE Duration**	**:**	3 Hours

Prerequisites:	
Courses	Artificial Intelligence / Machine Learning Fundamentals, Internet of Things , Data Structures and Algorithms
Knowledge on Fundamental Concepts	Basic knowledge of Python programming, machine learning, IoT, data handling and computer systems.
Knowledge on Connected Tools (If any)	Python, Jupyter Notebook, MQTT, Node-RED, MATLAB/Simulink or equivalent Digital Twin/IoT platforms.

Course Learning Objectives: This course will enable the students to	
CLO1	Explain the fundamental concepts, characteristics, architectures and lifecycle of Digital Twin systems.
CLO2	Apply IoT, sensing, data acquisition, modelling and synchronization techniques for developing Digital Twin systems.
CLO3	Apply Artificial Intelligence and Machine Learning techniques for prediction, anomaly detection, fault diagnosis and intelligent monitoring in Digital Twins.
CLO4	Analyze simulation, optimization and decision-making techniques for developing intelligent and adaptive Digital Twin systems.
CLO5	Explore AI-enabled Digital Twin applications, emerging technologies, security, trustworthiness and future trends.

Module – 1	**08 Hrs**
Introduction to Digital Twins: Introduction to Digital Twins: Need for Digital Twins, evolution of Digital Twin technology, characteristics of Digital Twins, Digital Twin versus Digital Model and Digital Shadow, physical and virtual entities, synchronization and fidelity. **Digital Twin Architecture:** Digital Twin components, Digital Twin ecosystem, Digital Twin lifecycle, Digital Twin architecture, physical-to-digital and digital-to-physical interaction, Digital Thread, Digital Twin systems and system-of-systems. **Types of Digital Twins:** Product Twin, Process Twin, Asset Twin, System Twin and Enterprise Twin.	
Module – 2	**08 Hrs**
Data and IoT for Digital Twins: Role of IoT in Digital Twin systems, sensors and actuators, data acquisition, real-time and historical data, IoT data pipelines, edge, fog and cloud computing. **Communication and Integration:** Communication protocols: MQTT, HTTP and OPC-UA. Data integration, interoperability, APIs and data exchange. **Digital Twin Modelling:** Physics-based modelling, data-driven modelling, simulation-based modelling and hybrid modelling. **Data Processing:** Data preprocessing, data cleaning, feature extraction, feature engineering, time-series data, data synchronization and contextualization. **Digital Twin Platform:** IT/OT infrastructure, virtual representation, integration services, synchronization services, analytics and application layers.	
Module – 3	**08 Hrs**

Artificial Intelligence for Digital Twins: Role of AI in Digital Twin systems, AI/ML pipeline, supervised learning, unsupervised learning and reinforcement learning.
Predictive Analytics: Regression, classification and time-series forecasting for Digital Twin applications.
Intelligent Monitoring: Anomaly detection, fault detection, fault diagnosis, condition monitoring and state estimation.
Predictive Maintenance: Predictive maintenance concepts, equipment health monitoring, remaining useful life prediction, failure prediction and maintenance decision support.
Deep Learning for Digital Twins: Artificial Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, LSTM networks and Autoencoders.

Module – 4	**08 Hrs**

Simulation and AI-Driven Decision Making: Simulation in Digital Twins, real-time simulation, what-if analysis, scenario analysis and virtual experimentation.
AI-Based Modelling: Surrogate models, reduced-order models, data-driven dynamical systems and hybrid AI-physics models.
Optimization: Optimization in Digital Twins, single-objective and multi-objective optimization, constraint-based optimization and Pareto-based decision-making.
Reinforcement Learning: Fundamentals of reinforcement learning, agent, environment, reward and policy, application of reinforcement learning to Digital Twin optimization and control.
Adaptive Digital Twins: Continuous learning, model updating, adaptive Digital Twins and closed-loop intelligence.

Module – 5	**08 Hrs**

AI-Enabled Digital Twin Applications: Manufacturing, Automotive, Healthcare, Smart Cities and Infrastructure, Energy.
Trustworthy Digital Twins
Security and privacy, cybersecurity, data integrity, model reliability, interoperability, explainability and human-in-the-loop decision-making.
Emerging Trends
Generative AI and Digital Twins, AI agents, multi-agent Digital Twin systems, edge AI, autonomous Digital Twins and Digital Twin maturity.

Topics for Self-learning	
Module – 1	Digital Twin maturity levels, Digital Twin taxonomy and Digital Twin use cases.
Module – 2	Digital Twin data standards, OPC UA, MQTT, edge computing, and cloud-based Digital Twin platforms.
Module – 3	SHAP-based explainable AI, Autoencoders for anomaly detection and remaining useful life prediction.
Module – 4	Digital Twin-based reinforcement learning, surrogate modelling and Physics-Informed Neural Networks.
Module – 5	Generative AI for Digital Twins, AI agents, multi-agent Digital Twin systems and autonomous Digital Twins.

Course Outcomes: After completing the course, the students will be able to	
CO1	Outline the fundamental concepts, architectures, lifecycle and applications of Digital Twin systems.
CO2	Apply IoT, data acquisition, modelling and synchronization techniques for developing Digital Twin systems.
CO3	Apply suitable AI and Machine Learning techniques for prediction, anomaly detection and intelligent monitoring in Digital Twins.
CO4	Analyze simulation, optimization and AI-based decision-making techniques for intelligent Digital Twin systems.
CO5	Examine AI-enabled Digital Twin applications, security, trustworthiness and emerging technologies.

Text Books	
1	**Aurélien Géron**, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 3rd Edition, O'Reilly Media, 2022, ISBN 978-1098125974.
2	**Richard S. Sutton and Andrew G. Barto**, "Reinforcement Learning: An Introduction", 2nd Edition, MIT Press, 2018, ISBN 978-0262039246.
3	**Samuel Greengard**, "The Internet of Things", Revised and Updated Edition, MIT Press, 2021, ISBN 978-0262542623.

Reference Books	
1	**Ian Goodfellow, Yoshua Bengio and Aaron Courville**, "Deep Learning", 1st Edition, MIT Press, 2016, ISBN 978-0262035613.
2	**S. S. Rao**, "Engineering Optimization: Theory and Practice", 5th Edition, Wiley, 2019, ISBN 978-1119454816.
3	**Jerry Banks, John S. Carson II, Barry L. Nelson and David M. Nicol**, "Discrete-Event System Simulation", 5th Edition, Pearson, 2010, ISBN 978-0136062127.

Web links and Video Lectures (e – Resources)	
1	https://www.digitaltwinconsortium.org/?utm_source=chatgpt.com
2	https://www.nvidia.com/en-us/omniverse/?utm_source=chatgpt.com
3	https://learn.microsoft.com/en-us/azure/digital-twins/?utm_source=chatgpt.com

Semester: II						
Minor Project /Ability Skill Development Project						
Course Code	:	1MSCS208		**Credits**	:	02
Hours/Week(L:T:P)	:	3		**CIE**	:	50 Marks
Total Hours	:	60		**SEE**	:	50 Marks
Course Type	:	PCC		**SEE Duration**	:	3 Hours

Course Learning Objectives: This course will enable the students to	
CLO1	To support independent learning and innovative attitude.
CLO2	To guide to select and utilize adequate information from varied resources upholding ethics.
CLO3	To guide to organize the work in the appropriate manner and present information (acknowledging the sources) clearly.
CLO4	To develop interactive, communication, organization, time management, and presentation skills.
CLO5	To impart flexibility and adaptability.
CLO6	To inspire independent and team working.
CLO7	To expand intellectual capacity, credibility, judgments, intuition.
CLO8	To adhere to punctuality, setting and meeting deadlines.
CLO9	To instill responsibilities to oneself and others.
CLO10	To train students to present the topic of project work in a seminar without any fear, face audience confidently, enhance communication skill, involve in group discussion to present and exchange ideas.

Course Outcomes: After completing the course, the students will be able to	
CO1	Present the mini-project and be able to defend it.
CO2	Make links across different areas of knowledge and to generate, develop and evaluate ideas an d information so as to apply these skills to the project task.
CO3	Habituated to critical thinking and use problem solving skills.
CO4	Communicate effectively and to present ideas clearly and coherently in both the written and ora l forms.
CO5	Work in a team to achieve common goal.
CO6	Learn on their own, reflect on their learning and take appropriate actions to improve it.

Mini-Project	
1	Each student of the project batch shall involve in carrying out the project work jointly in constant consultation with internal guide, co-guide, and external guide and prepare the project report as per the norms avoiding plagiarism.

Semester: II						
Computer Networks Lab						
Course Code	:	1MSCSL207		Credits	:	01
Hours/Week (L:T:P:SL)	:	0:0:28:02		CIE	:	50 Marks
Total Hours	:	30		SEE	:	50 Marks
Course Type	:	PCC (AEC Lab)		SEE Duration	:	3 Hours

Prerequisites:	
Courses	Data communication, Computer Networks, Programming languages (C/java).
Knowledge on Fundamental Concepts	OSI and TCP/IP models, IP addressing and subnetting, TCP and UDP etc.,
Knowledge on Connected Tools (If any)	Software Tools Cisco Packet Tracer, NS-3 and Wireshark.

Course Learning Objectives: This course will enable the students to	
CLO1	Implement basic networking algorithms and protocols using suitable programs.
CLO2	Configure and verify computer networks using Cisco Packet Tracer.
CLO3	Develop basic TCP/UDP socket programs and analyze network traffic using Wireshark.

Sl.No.	Experiments
	Study experiment
	a. Understand basics of Network models (layering, Protocols, etc.,) b. Installing and Understand of CISCO Packet Tracer and Wireshark
	Practical Experiment
1	Develop a program for error detection using CRC-CCITT (16-bits).
2	Develop a program to implement Sliding Window Protocol
3	Develop a program to find shortest path using Bellman ford Routing algorithm
4	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
5	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
6	Setup simple LAN in packet tracer using PCs, switches, and routers.
7	Configure routers using static routes for inter-network communication.
8	Implement Network Address Translation (NAT) and Access Control Lists (ACLs), demonstrating how to permit internal users internet access while blocking unauthorized external traffic.
9	Implementation and Verification of IPv6 Addressing, Stateless address auto configuration (SLAAC), and Static Routing.
10	Tool Exploration –Wireshark : a) Observe TCP Three-Way Handshake using Wireshark. b) Packet Capture and Protocol Analysis

Self-Learning : Sample list
• VLAN and Inter-VLAN Routing • OSPF Routing • DHCP and DNS Configuration • NAT and ACL Configuration

Course Outcomes: After completing the course, the students will be able to	
CO1	Implement basic networking algorithms and socket programs.
CO2	Configure and verify IPv4/IPv6 networks using Packet Tracer.
CO3	Analyze network communication using Wireshark and network utilities.

Institute Vision

Preparing Competent Engineering and Management Professionals to Serve the Society

Institute Mission

- Providing Students with a Sound Knowledge in Fundamentals of their branch of Study.
- Promoting Excellence in Teaching, Training, Research and Consultancy.
- Exposing Students to Emerging Frontiers in various domains enabling Continuous Learning.
- Developing Entrepreneurial acumen to venture into Innovative areas.
- Imparting Value based Professional Education with a sense of Social Responsibility.

Department Vision

Building Technically Competent Engineers to Serve the Society

Department Mission

- Imparting strong fundamental knowledge in Mechanical Engineering and allied fields.
- Fostering a culture of collaboration and propagating Industry institute interactions to enhance advances in latest technologies.
- Creating State-of-Art facilities benefitting students and faculty to carryout Research, Innovation and Consultancy.
- Providing conducive environment for both students and faculty to engage in lifelong learning activities.
- Inculcating Professional ethics, Leadership qualities and Entrepreneurial skills among students.

SJC INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU from 2024-25

P.B. No.20, B.B Road,

Chikkaballapur - 562 101, Karnataka