

M. TECH – COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

The **M.Tech. in Computer Science and Engineering** is offered by the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham. The programme is designed to prepare graduates with strong theoretical foundations, advanced technical expertise, and research capabilities to address the challenges of modern computing.

Computer Science and Engineering is a rapidly evolving discipline that drives innovation across industries. Emerging technologies such as Artificial Intelligence, Cloud Computing, Data Science, Internet of Things (IoT), Quantum Computing, and Cyber Security are built upon strong foundations in algorithms, programming, computing systems, mathematics, and software engineering. This programme equips students with the knowledge and skills required to design, develop, and deploy intelligent and scalable computing solutions for real-world applications. The curriculum provides a balanced combination of core and advanced courses covering mathematics, programming, algorithms, software engineering, computer systems, networks, and emerging technologies. Students can further specialize through a wide range of electives in areas such as Artificial Intelligence, Machine Learning, Data Science, Internet of Things, Edge, Fog and Cloud Computing, Cyber Security, Mobile Computing, Full Stack Engineering, Quantum Computing, and related domains. The programme emphasizes hands-on learning, project-based education, research, and industry-relevant skill development to prepare graduates for both academia and industry.

A significant component of the programme is the research thesis, which enables students to apply advanced concepts to solve challenging research problems under faculty guidance. The curriculum is designed to foster analytical thinking, innovation, and independent research while preparing students for emerging technological trends and interdisciplinary applications. The programme is suitable for graduates with a Bachelor's degree in Computer Science, Information Technology, or other computing-related disciplines. It also provides an opportunity for graduates from allied disciplines to strengthen their expertise in advanced computing and pursue specialized careers or research.

Graduates of the programme are well prepared for diverse career opportunities, including Software Engineer, Software Architect, Full Stack Developer, Technical Architect, AI Engineer, Machine Learning Engineer, Data Scientist, Big Data Engineer, Cloud Solutions Architect, Cyber Security Analyst, Research Engineer, and Computer Network Architect. They are also equipped to pursue doctoral studies and advanced research in premier academic and industrial research organizations.

To enhance industry readiness, students have opportunities to undertake internships at leading industries and research laboratories for periods of up to one year. The programme also offers opportunities for international academic exposure through student exchange and dual-degree collaborations with partner universities, including Vrije Universiteit Amsterdam (Netherlands), the University of California, Davis (USA), and the University of New Mexico (USA). Students interested in research careers are encouraged to continue through the Integrated Ph.D. programme at Amrita Vishwa Vidyapeetham.

Program Educational Objectives (PEO)

1. Demonstrate application and adaptation of core concepts in Computer Science and Engineering in industry or research and become prolific professionals and entrepreneurs.
2. Pursue lifelong learning to adapt to emerging computing trends and design computing solutions for real world inter-disciplinary problems.
3. Demonstrate high regard for professionalism, team-spirit, integrity and respect diversity, societal needs and sustainability when designing technological solutions.

Program Outcomes (PO)

At the end of the M. Tech CSE programme the students will be

1. Able to demonstrate a mastery over the foundations of Computer Science and Engineering specifically with respect to solving problems, designing algorithms and systems, and upcoming key technologies
2. Able to design and develop computing solutions using emerging computing paradigms to interdisciplinary problems following standard practices, tools and technologies
3. Able to demonstrate independent study and life-long learning in order to adapt to the changing landscape of technology and computing trends
4. Able to independently carry out research investigation and development work to solve practical problems
5. Able to write and present a substantial technical report/document
6. Able to demonstrate commitment to professional ethics

Ideas behind the design of POs

1. Program Outcomes PO1, PO4 and PO5 have been adopted from NBA (Ref. https://www.nbaind.org/files/PG_Eng_Annexure/PG_Engineering_Manual.pdf Page 16). PO1 has been elaborated to suit our M. Tech CSE.
2. PO1 is about foundational knowledge and skill (and has relevance to all courses). PO2 is about state-of-the-art and current. PO3 is futuristic. PO2 and PO3 very well relate to case studies, lab implementations and dissertation.
3. PO4 and PO5 (adopted from NBA) relate to dissertation as well as to case studies.
4. PO6 explicitly mentions professional ethics. Ethics and integrity have been part of our mission statements and PEO. PO6 aligns with those explicitly. There are various facets of ethics and integrity which we can map for our courses.

M.Tech Computer Science & Engineering

CURRICULUM

I Semester

Course Code	Type	Subject	L T P	Credits
26CS601	FC	Advanced-Data Structures and Algorithms	3-0-2	4
26MA602	FC	Mathematical Foundations for Computer Science & Engineering	3-1-0	4
	SC	Soft Core 1	3-0-2	4
	SC	Soft Core 2	3-0-2	4
26RM603	FC	Research Methodology	2-0-0	2
	HU	Amrita Values Program*		P/F
25AVP501	HU	Mastery Over Mind	1-0-2	2
26CDS101	HU	Career Development Skills I	0 0 3	P/F
		Credits		20

II Semester

Course Code	Type	Subject	L T P	Credits
	SC	Soft Core 3	3-0-2	4
	SC	Soft Core 4	3-0-2	4
	SC	Soft Core 5	3-0-2	4
	E	Elective 1	2-0-2	3
	E	Elective 2	2-0-2	3
26CDS611	HU	Career Development Skills II	0 0 3	1
26CUL500	HU	Integrated Amrita Meditation Technique	0-0-2	1
		Credits		20

III Semester

Course Code	Type	Subject	L T P	Credits
	E	Elective 3	2-0-2	3
	E	Elective 4	2-0-2	3
26CS798	P	Dissertation-I		10
		Credits		16

IV Semester

Course Code	Type	Subject	L T P	Credits
26CS799	P	Dissertation-II		16

Total Credits 72

List of Courses

FOUNDATION CORE

Course Code	Subject	L T P	Credits
26CS601	Advanced Data Structures and Algorithms	3-0-2	4
26MA602	Mathematical Foundations for Computer Science & Engineering	3-1-0	4
26RM603	Research Methodology	2-0-0	2

SOFT CORE

Course Code	Subject	L T P	Credits
26CS631	Software Engineering with Agile and DevOps	3-0-2	4
26CS632	Distributed Systems	3-0-2	4
26CS633	Advanced Networks	3-0-2	4
26CS634	Foundations of Cyber Security	3-0-2	4
26CS635	Full Stack Development	3-0-2	4
26CS636	Modern Database Management Systems	3-0-2	4
26CS637	Machine Learning	3-0-2	4
26CS638	Foundations of AI	3-0-2	4

OPEN ELECTIVES

Students may choose elective courses offered by other M.Tech. programmes, such as Artificial Intelligence, Data Science, Cyber Security, and related specializations, subject to eligibility and approval.

ELECTIVES

Course Code	Subject	L T P	Credits
26CS731	Quantum Computing	2-0-2	3
26CS732	Mobile Application Development	2-0-2	3
26CS733	Software Quality Assurance and Testing	2-0-2	3
26CS734	Fog and Edge Computing	2-0-2	3
26CS735	Augmented Reality and Virtual Reality	2-0-2	3
26CS736	Block chain Technology	2-0-2	3
26CS737	Cloud Computing and Services	2-0-2	3

26CS738	Deep Learning	2-0-2	3
26CS739	Internet of Things	2-0-2	3
26CS740	Wireless and Mobile Networks	2-0-2	3
26CS741	Software Defined Systems	2-0-2	3
26CS742	Big Data Analytics	2-0-2	3
26CS743	High Performance Computing	2-0-2	3
26CS744	Generative AI and Large Language Models	2-0-2	3
26CS745	AI Ethics and Responsible AI	2-0-2	3
26CS746	Deep Reinforcement Learning	2-0-2	3
26CS747	Mobile Network Technologies - 5G and Beyond	2-0-2	3
26CS748	Computer Vision	2-0-2	3
26CS749	Knowledge Systems and Semantic AI	2-0-2	3
26CS750	Evolving Intelligence	2-0-2	3

SYLLABUS

FOUNDATION CORE

26CS601

ADVANCED DATA STRUCTURES AND ALGORITHMS

3-0-2-4

Pre-Requisite(s): Foundations of Data Structures and Algorithms, Programming Paradigms and Problem Solving

Course Type: Lab

Course Objectives

- Gain an in-depth knowledge of advanced data structures viz., hash tables, bloom filters and skip lists
- Improve the efficiency of algorithms for different data structures to enhance performance in real-world applications.
- Strengthen critical thinking and problem-solving skills by analyzing complex problems and devising efficient solutions using various algorithmic techniques.

Course Outcomes

CO1	Comprehend the theoretical foundations of algorithm analysis and analyze the complexity of data structures and algorithms.
CO2	Apply advanced data structures for real-world problem-solving.
CO3	Employ various algorithm design techniques for solving real-world problems.
CO4	Evaluate the complexity classes of various problems and relate them to classical problems.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1				
CO2	3	2	1	1		1
CO3	3	2	1	1		1
CO4	3	2	1	1		

Syllabus

Algorithm Analysis: Methodologies for analyzing algorithms, asymptotic growth rates, and amortized analysis. Fundamental Data Structures: Array-based structures and linked lists. Advanced Data Structures: Dictionaries, hash tables, Bloom filters, binary search trees, interval trees, range trees, and skip lists.

Algorithm Design Techniques: Foundations and applications of Divide-and-Conquer, Greedy techniques, Dynamic Programming, Backtracking, and Branch and Bound. Graph Algorithms: Topological sorting, strongly connected components, bi-connected components, bridges, articulation points, single-source shortest paths, and all-pairs shortest paths.

Flow Networks: Ford–Fulkerson algorithm, Edmonds–Karp algorithm, and applications of maximum flow, including efficient algorithms for maximum bipartite matching. Computational Complexity: NP-completeness, important NP-complete problems, polynomial-time reductions, and approximation algorithms. Parallel Algorithms (Overview): Tree contraction, parallel Divide-and-Conquer, and maximal independent set algorithms.

Text Books/References

1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser, John Wiley & Sons, 2013.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 3rd ed., Prentice Hall of India, 2009.
3. Algorithm Design and Applications, Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons, 2014.
4. Randomized Algorithms, Rajeev Motwani and Prabhakar Raghavan, Cambridge University Press, 1995.
5. Introduction to Algorithms: A Creative Approach, Udi Manber, Addison-Wesley, 1989.
6. Approximation Algorithms, Vijay V. Vazirani, Springer, 2003.
7. The Algorithm Design Manual, Steven S. Skiena, 3rd ed., Springer, 2020.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, programming assignments, algorithm design and analysis exercises, case studies, laboratory assessments, and project reviews. The Mid-Term examination and the End-Semester examination may be conducted as a two-hour written examination or a laboratory-integrated examination, based on the course requirements.

Pre-Requisite(s): Basics of Linear Algebra

Course Type: Theory

Course Objectives

- This course aims to provide solid mathematical foundations to understand, formulate and solve important problems in real-life context in the field of Computer Science and Engineering.
- The foundational knowledge offered in this course will also provide necessary skills to quantify and analyze the existing computational systems thus developing an in-depth understanding of such systems.

Course Outcomes

CO1	Understand and apply the basic concepts of vector spaces, subspaces, linear independence, span, basis and dimension and analyze such properties on the given set.
CO2	Understand and apply the concept of inner products and apply it to define the notion of length, distance, angle, orthogonality, orthogonal complement, orthogonal projection, orthonormalization and apply these ideas to obtain least square solution.
CO3	Understand the theory of random variables and distributions to analyze the data.
CO4	Understand the theory of two random variables and analyze the relationship in data analytics.
CO5	Understand the statistical procedure of hypothesis testing and use it to analyze the data.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	3		1		
CO2	2	2		2		
CO3	1	2		2		
CO4	1	2		2		
CO5	1	2		2		

Syllabus

Vectors Spaces: Vector spaces, basis and dimension, change of basis, orthogonality, and the Gram–Schmidt process. Fundamental Subspaces: Column space, null space, row space, and left null space. Matrix Decompositions and Applications: Projections, least squares, and linear regression; eigenvalue decomposition and diagonalization; special matrices, similarity transformations, algorithms, and singular value decomposition (SVD).

Probability Theory: Probability models and axioms, conditional probability, independence, and Bayes' rule. Random Variables: Discrete random variables, probability mass functions (PMFs), expectations, and joint PMFs; continuous random variables, probability density functions (PDFs),

expectations, joint PDFs, and continuous Bayes' rule. Probability Distributions: Binomial, Poisson, geometric, exponential, uniform, and normal distributions. Statistical Concepts: Derived distributions, convolution, covariance, correlation, the weak law of large numbers, and the central limit theorem.

Statistical Inference: Parameter estimation, hypothesis testing, and applications of hypothesis testing through case studies. Regression and Analysis: Regression analysis and analysis of variance (ANOVA). Non-Parametric Methods and Experimental Design: Non-Parametric hypothesis tests and design of experiments.

Text Books/References

1. Linear Algebra for Everyone, Gilbert Strang, Wellesley-Cambridge Press, 2020.
2. Linear Algebra Done Right, Sheldon Axler, Springer Nature, 2024.
3. Elementary Linear Algebra, Howard Anton and Chris Rorres, 10th ed., John Wiley & Sons, 2010.
4. Probability and Statistics for Computer Science, David Forsyth, Springer International Publishing, 2018.
5. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, CRC Press, 2012.

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Minimum 5 Components)	50	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, assignments, problem-solving exercises, case studies, laboratory/computational exercises (where applicable), and project reviews. The Mid-Term and End-Semester examinations may be conducted as two-hour written examinations or as laboratory-integrated examinations, as appropriate for the course, subject to the approval of the Department.

Pre-Requisite(s): None

Course Type: Theory/Case Study

Course Objectives

- To enable students to define research problems, review, analyze as well as to evaluate literature and possibly to formulate effective solutions.
- To prepare students either for a research thesis or for an industry-based project.
- To provide oral and written communication skills.
- To inculcate a strict adherence to the principles of research ethics and values.

Course Outcomes

CO1	Understand the basic concepts of research and its methodologies.
CO2	Understand and apply the process of searching for, selecting and critically analyzing research articles and papers.
CO3	Formulate and evaluate research questions and apply the process of designing a research study and interpreting the outcomes of the study.
CO4	Write and present a research report and thesis.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1		1	1		
CO2		1	2	2		1
CO3	2	2		2	3	2
CO4				2		2

Syllabus

Meaning of Research, Types of Research, Research Process, Problem definition, Objectives of Research, Research Questions, Research design, Approaches to Research - Quantitative vs. Qualitative Approach, Understanding Theory, Building and Validating Theoretical Models, Exploratory vs. Confirmatory Research, Experimental vs Theoretical Research, Importance of reasoning in research.

Problem Formulation, Understanding Modeling & Simulation, Conducting Literature Review, Referencing, Information Sources, Information Retrieval, Role of Libraries in Information Retrieval, Tools for identifying literatures, Indexing and abstracting services, Citation indexes.

Experimental Research: Cause effect relationship, Development of Hypothesis, Measurement Systems Analysis, Error Propagation, Validity of experiments, Statistical Design of Experiments, Field Experiments, Data/Variable Types & Classification, Data collection, Numerical and Graphical Data Analysis: Sampling, Observation, Surveys, Inferential Statistics, and Interpretation of Results

Preparation of Dissertation and Research Papers, Guidelines for writing the abstract, introduction, methodology, results and discussion, conclusion sections of a manuscript. References, Tables and illustrations and Citation.

Intellectual property rights (IPR) – patents – copyrights - Trademarks - Ethics of Research- Scientific Misconduct - Forms of Scientific Misconduct – Plagiarism - Unscientific practices in thesis work.

Text Books/References

1. Research Design and Methods: A Process Approach, Kenneth S. Bordens and Bruce B. Abbott, 11th ed., McGraw-Hill Education, 2022.
2. Statistical Research Methods: A Guide for Non-Statisticians, Roy Sabo and Edward Boone, Springer, 2013.
3. Scientific Papers and Presentations, Michael Davis, Karen Davis, and Marion Dunagan, 3rd ed., Elsevier Inc., 2013.
4. Handbook of Research Ethics and Scientific Integrity, Ron Iphofen (Ed.), Springer, 2020.
5. Elsevier – Ethics in Research & Publication, Elsevier

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Minimum 5 components)	50	
End Semester		30

Note: Continuous assessment may include quizzes, assignments, literature reviews, research proposal development, case studies, seminar presentations, project reviews, and other research-oriented evaluation components. End-Semester assessments may be conducted as a two-hour written examination and/or through the evaluation of research-based components such as a research proposal, literature survey, case study analysis, presentation, or viva voce, as prescribed for the course.

Course Type: Project

Course Objectives

- The student is expected to carry out supervised research resulting in comprehensive analytical study of the area of interest.
- The student should be able to conduct an intensive review of literature.
- Analysis of existing work through comprehensive experimental analysis resulting in identification of a novel and well-defined problem

Course Outcomes

CO1	Demonstrate sound fundamentals in a chosen area of computing.
CO2	Identify and formulate research problems through conduct of an intensive review of literature.
CO3	Analyze existing work solving the defined problem through comprehensive experimental analysis.
CO4	Effectively communicate the work at all stages of the project adhering to ethical practices.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	2	2	2
CO2	2	3	3	3	2	3
CO3	3	3	3	2	2	3
CO4	2	3	3	2	2	3

References

1. Relevant literature for the computing problem.

Evaluation: 80/20

Note: The Dissertation shall be carried out under the supervision of a faculty member approved by the Department. Students are expected to identify a research problem, conduct independent research, and submit a dissertation demonstrating originality, technical competence, and analytical rigor. The dissertation shall be evaluated based on periodic progress reviews, the quality of the dissertation report, and the final presentation and viva voce examination.

Pre-Requisite(s): Dissertation Phase I

Course Type: Project

Course Objectives

- The course shall be pursued for a minimum of 16 weeks during the final semester, following the preliminary work carried out in Phase-1 during the previous semester.
- The student is expected to enhance the knowledge base in the chosen field of research in computing.
- The student demonstrates the ability to design effective and novel solutions to the defined problem in Phase 1 and through rigorous experimental and theoretical analysis demonstrates the effectiveness of the proposed solution.
- The student is also expected to effectively communicate the scholarly outcomes such as presentations, report, and publish them in a reputed conference or journal.

Course Outcomes

CO1	Identify gaps and needs in the chosen areas to refine the problem defined.
CO2	Design and develop novel and efficient solutions to the problem and analyze results.
CO3	Prepare the thesis report and defend the thesis based on the work done.
CO4	Augment the knowledge base in the chosen area of computing by publishing scholarly articles and adhere to ethical practices at every stage.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	3	3	2
CO2	3	3	3	3	2	3
CO3	3	3	3	2	2	3
CO4	3	3	3	3	3	3

References

1. Relevant literature for the computing problem.

Evaluation: 80/20

Note: The Dissertation shall be carried out under the supervision of a faculty member approved by the Department. Students are expected to identify a research problem, conduct independent research, and submit a dissertation demonstrating originality, technical competence, and analytical rigor. The dissertation shall be evaluated based on periodic progress reviews, the quality of the dissertation report, and the final presentation and viva voce examination.

SOFT CORE

26CS631

SOFTWARE ENGINEERING WITH AGILE AND DEVOPS

3-0-2-4

Pre-Requisite(s): Software Engineering

Course Type: Project

Course Objectives

- To introduce students to industry standard agile practices and Devops
- To explore a variety of applications using Agile methodology

Course Outcomes

CO1	Explain Agile principles, frameworks, and project management practices, and compare them with traditional software development approaches.
CO2	Apply Agile requirement engineering, planning, estimation techniques, and Scrum practices to manage software projects.
CO3	Apply Agile testing, quality engineering, and test automation techniques using modern testing tools and frameworks.
CO4	Develop CI/CD pipelines and deploy applications using DevOps tools, containerization, and cloud-native technologies.
CO5	Evaluate modern software delivery practices including DevSecOps, GitOps, observability, and AI-assisted software development.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1				
CO2	1	1			3	
CO3	3	3	1	1		2
CO4	3	3	1		1	2
CO5	3	3			1	2

Syllabus

Agile Software Development and Project Management: Introduction to Agile software development, Agile values and principles, comparison of Agile and traditional software development models, Agile frameworks including Scrum, Kanban, XP, Lean, SAFe, Scrum@Scale, and Disciplined Agile. Agile requirement engineering using personas, user stories, story mapping, 3Cs, INVEST principle, acceptance criteria, product backlog, backlog refinement, Definition of Ready (DoR), Definition of Done (DoD), Agile estimation techniques, sprint planning, velocity, forecasting, product roadmaps, burndown and burnup charts, information radiators, flow metrics, and Agile project management tools.

Agile Quality Engineering and Testing: Agile quality engineering, verification and validation, Agile testing principles, testing quadrants, Shift Left and Shift Right testing, TDD, BDD, ATDD, unit, integration, functional, API, UI, regression, performance, security, and accessibility testing. Test automation pyramid, automation frameworks, Selenium, Cypress, Playwright, Postman, quality metrics, defect management, and AI-assisted test automation.

DevOps and Cloud-Native Software Delivery: DevOps principles and culture, Continuous Integration and Continuous Delivery/Deployment (CI/CD), Git workflows, Jenkins, GitHub Actions, Docker, Kubernetes, Infrastructure as Code (IaC), cloud-native application development, continuous monitoring and observability, DevSecOps, GitOps, Platform Engineering, software reliability, AI-assisted DevOps, Generative AI in software engineering, modern software delivery practices, and Introduction to MLOps: machine learning lifecycle, model deployment, monitoring, and CI/CD for machine learning.

Text Books/References

1. Essential Scrum: A Practical Guide to the Most Popular Agile Process, Kenneth S. Rubin, Addison-Wesley Professional, 2012.
2. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Jez Humble and David Farley, Addison-Wesley Professional, 2011.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim, Patrick Debois, John Willis, and Jez Humble, 2nd ed., IT Revolution Press, 2021.
4. Learning Agile: Understanding Scrum, XP, Lean, and Kanban, Andrew Stellman and Jennifer Greene, O'Reilly Media, 2015.
5. Agile Testing: A Practical Guide for Testers and Agile Teams, Lisa Crispin and Janet Gregory, Addison-Wesley Professional, 2009.
6. User Stories Applied: For Agile Software Development, Mike Cohn, Addison-Wesley Professional, 2004.
7. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, Nicole Forsgren, Jez Humble, and Gene Kim, IT Revolution Press, 2018.
8. Automation in Testing: Modern Strategies to Test Faster, Mark Winteringham, Manning Publications, 2022.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): None

Course Type: Lab

Course Objectives

- Understand the design of distributed systems and algorithms that support distributed computing.
- Gain practical exposure to the design and functioning of existing distributed systems and algorithms.

Course Outcomes

CO1	Understand the design principles in distributed systems and the architectures for distributed systems.
CO2	Apply various distributed algorithms related to clock synchronization, concurrency control, deadlock detection, and termination.
CO3	Analyze techniques for Consistency, fault tolerance and recovery in distributed systems.
CO4	Analyze the design and functioning of existing distributed systems and file systems.
CO5	Implement different distributed algorithms over current distributed platforms.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	1			
CO2	3	3	1	1		
CO3	3	2				
CO4	3	2	1	1	1	
CO5	3	2	1	1	1	2

Syllabus

Introduction to Distributed Systems: Characteristics and challenges; Types of distributed systems: Client–Server, Peer-to-Peer (P2P), Cluster, Grid, Cloud, Edge, Fog, and Cyber-Physical Systems (CPS); Distributed system architecture, process overview, taxonomy of distributed systems, scalability, performance, load balancing, and availability; Models of computation: Shared memory and message-passing systems, synchronous and asynchronous systems; Communication in distributed systems: Remote Procedure Calls (RPC), REST APIs, gRPC, message-oriented communication, publish–subscribe, and event-driven communication.

Logical time and event ordering: Happened-Before Relation, Lamport Logical Clocks, Vector Clocks, Causal Ordering, Total Ordering; Physical clock synchronization: Clock Drift, Clock Skew, Network Time Protocol (NTP); Global state and snapshot algorithms: Consistent Global State, Chandy–Lamport Snapshot Algorithm; Distributed mutual exclusion: Centralized Algorithm,

Ricart–Agrawala Algorithm, Token Ring Algorithm; Leader election: Bully Algorithm, Ring Algorithm; Distributed deadlock detection and termination detection; Distributed consensus: Consensus Problem, Introduction to Paxos, Raft Algorithm, Byzantine Fault Tolerance (BFT); Distributed transactions: Two-Phase Commit (2PC), Three-Phase Commit (3PC); Coordination services: Introduction to ZooKeeper and distributed coordination.

Distributed database architecture and design; Distributed file systems: Dropbox, Google File System (GFS), Hadoop Distributed File System (HDFS); Distributed data processing: MapReduce programming model, Apache Spark, and Resilient Distributed Datasets (RDDs); Distributed storage models: Introduction to NoSQL databases, Bigtable, and HBase; Consistency control: Data-centric consistency, Client-centric consistency, CAP Theorem, Replica Management; Fault tolerance and recovery: Basic concepts, fault models, recovery mechanisms, and checkpointing.

Case Studies / Project: Analysis of large-scale distributed platforms such as Google Spanner, Apache Spark, Amazon Aurora, Distributed File Systems (GFS/HDFS), Edge Computing Platforms, Cyber-Physical Systems, and Blockchain-based Distributed Systems.

Laboratory: MPI/OpenMP, Apache Spark.

Text Books/References

1. Distributed Systems, Andrew S. Tanenbaum and Maarten van Steen, 3rd ed., Pearson, 2017.
2. Distributed Computing: Principles, Algorithms, and Systems, Ajay D. Kshemkalyani and Mukesh Singhal, Cambridge University Press, 2012.
3. Elements of Distributed Computing, Vijay K. Garg, John Wiley & Sons, 2002.
4. Distributed Systems: Concepts and Design, George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, 5th ed., Pearson Education, 2017.
5. Distributed Algorithms: An Intuitive Approach, Wan Fokkink, 2nd ed., MIT Press, 2018.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Computer Networks

Course Type: Lab

Course Objectives

- Deep understanding of TCP / IP and L2 protocols in high-speed networks.
- Proper usage of various enabling technologies and protocols related to wireless and mobile networks for practical applications.
- Application and usage of various network services in wireless intelligent networks as well as adoption of SDN and NFV for IoT applications.

Course Outcomes

CO1	Analyze roles of TCP/IP protocol as well as MAC protocols in high-speed networks.
CO2	Apply the various enabling technologies and protocols related to wireless and mobile networks for practical applications.
CO3	Understand and apply various network services in Wireless Sensor Networks.
CO4	Understand the design principles in SDN and NFV for IoT and apply for practical use case.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			
CO2	2	2	2			
CO3	2	2	2	2		
CO4	2	2	2	2		

Syllabus

Internetworking: Introduction to DCN (Data Center Networking) and Programmable Networks, Network Architecture and Design Principles, Layering, Names and addresses. TCP/IP suite of protocols, TCP extensions for high-speed networks, multimedia networking applications, IPv6: IPv6 and Transition Mechanisms, Quality of Service (QoS), Network Management using SNMP, Introduction to Network Security and Zero Trust Architecture.

Recent trends in Wireless communication networks, 5G and Introduction to 6G Networks, wireless sensor networks, mobile ad-hoc networks, Vehicular Ad-hoc Networks (VANETs), Internet of Things (IoT) Networking, mobility-based protocols, information aggregation, information storage and query, localization services, Performance Issues and Security in Wireless and IoT Networks.

SDN: Data and Control Plane, OpenFlow Architecture and Controllers, Network Function Virtualization for IoT, Cloud Networking, Security in SDN and NFV. Case Study: Network management tools used at ICTS, AVVP, Coimbatore Campus, Network protocols along with network security for any one Industrial Use Case.

Text Books/References

1. Computer Networking: A Top-Down Approach, James F. Kurose and Keith W. Ross, 7th ed., Pearson Education, 2017.
2. Computer Networks and Internets, Douglas E. Comer, 6th ed., Pearson Education, 2015.
3. Software Defined Networks: A Comprehensive Approach, Paul Goransson, Chuck Black, and Timothy Culver, Morgan Kaufmann, 2016.
4. Technical papers on course-related topics from Institute of Electrical and Electronics Engineers journals and conferences, and relevant IEEE Standards documents.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): None

Course Type: Lab

Course Objectives

- Knowledge in encryption and decryption using private and public key cryptography, digital signatures, cryptographic hash functions.
- Understand and apply authentication and authorization mechanisms to protect OS and database.
- Understand the threats and vulnerabilities and need for securing resources and exercising privacy preservation.

Course Outcomes

CO1	Understand the Fundamentals of Cyber Security Domain.
CO2	Analyze and Apply Authentication and Authorization Techniques.
CO3	Acquire foundational skills for developing expertise in one or more sub-domains of cyber-security and perform security reviews and audits.
CO4	Identify insights on how to apply Cyber Security to secure operating systems and database design.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		2
CO2	3	2	2	2		
CO3	3	3	2	2		2
CO4	3	2	2	3		

Syllabus

Security Concepts: Confidentiality, Integrity, and Availability (CIA), Confusion vs Diffusion, Stream vs Block ciphers, Private-Key vs Public Key Cryptography, Cryptanalysis. Pseudorandom functions, Feistel Networks and Non Feistel Networks, Key Exchange, Public Key Encryption systems. Hash Functions, Message Digest, Signing and Verification, AES, DES, X509 certificates, SSL, TLS, HTTPS, Open SSL, SSH, Elliptic Curve Cryptography.

Authentication and Authorization: Identity and Access Management, Factors, Multi-factor, Kerberos, Role based Access Control. Operating System Security: Windows and Linux, SELinux, ACL and Active Directory. Protection System, Authorization, Security Analysis and Vulnerabilities, Security issues related to Internet, Intranet, Cloud Computing, Embedded System, Mobile System, Internet of Things and Wireless Networks.

Database Security: Securing different types of data, Database security, Data sanitization, Attacks on Data, SQL Injection, Buffer-flow, Privacy of Data. Implementing Cryptographic schemes on datasets.

Tools: Wireshark, Snort, NPM, Open VAS, Burp Suite, Nessus, Wazuh, Suricata.

Text Books/References

1. Cryptography and Network Security: Principles and Practice, William Stallings, 8th ed., Pearson, 2023.
2. Mastering Python for Networking and Security, José Manuel Ortega, Packt Publishing, 2018.
3. Database and Application Security: A Practitioner's Guide, R. Sarma Danturthi, Addison-Wesley Professional, 1st ed., 2024.
4. Cryptography and Security, T. R. Padmanabhan, C. K. Shyamala, and N. Harini, 1st ed., Wiley India, 2011.
5. Computer Security: Art and Science, Matt Bishop, 2nd ed., Addison-Wesley, 2012.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Programming and database fundamentals

Course Type: Project

Course Objectives

- To understand the core concepts of both the front-end and back-end programming course.
- To explore the latest web development technologies for application design
- To apply and understand the in-depth study of the various web development tools

Course Outcomes

CO1	Gain a foundational understanding of web technologies, client-server architecture, and set up a development environment.
CO2	Develop proficiency in HTML, CSS, and JavaScript basics, including DOM manipulation, ES6+ features, and asynchronous programming.
CO3	Learn to use React for building dynamic user interfaces, including components, state management, hooks, and routing.
CO4	Understand server-side technologies with Node.js and Express.js, manage databases with MongoDB, and implement secure user authentication.
CO5	Explore full stack project development and deploy applications using hosting services with comprehensive testing and debugging practices.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	1	1	1	1	
CO2	2	2	1	3	2	
CO3	3	2	1	2	1	1
CO4	3	2	3	3	1	1
CO5	3	3	2	1	1	2

Syllabus

Frontend Development - Introduction to Web Development - Overview of Web Technologies - Understanding Client-Server Architecture - Setting up Development Environment. HTML & CSS - HTML and CSS Basics - JavaScript Basics - ES6+ Features: Arrow Functions, Template Literals, Destructuring. Asynchronous JavaScript: Promises, Async/Await- JavaScript Libraries: jQuery, D3.js. Frontend Frameworks - Introduction to React: Components, State, Props- Hooks, Context API, React Router, Typescript.

Backend Development - Introduction to Backend Development - Overview of Server-Side Technologies. Node.js & Express.js - basics - Building Web Servers with Express.js - Middleware and Routing - Handling Requests and Responses. Database Management - Working with MongoDB: CRUD Operations - Joins, Indexes, Transactions - Authentication & Security. User Authentication:

JWT - HTTPS, Data Validation, Sanitization - Building RESTful APIs- Integrating Frontend with Backend.

Full Stack Integration and Hosting - Introduction to Hosting Services: Heroku, AWS, DigitalOcean
- Containerization Basics - Exploring Full Stack sample Project Development. Testing & Debugging
- Frontend Testing - Jest, React Testing Library. Backend Testing - Mocha, Chai.

Text Books/References

1. Web Design with HTML, CSS, JavaScript and jQuery, Jon Duckett, John Wiley & Sons, 2014.
2. Full-Stack JavaScript Development, Eric Bush, Apress, 2020.
3. Mastering Full Stack React Web Development, Tomasz Dyl, Kamil Przeorski, and Maciej Czarnecki, Packt Publishing, 2017.
4. JavaScript for Web Developers, Nicholas C. Zakas, Wiley Publishing, 2012.
5. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites, Robin Nixon, O'Reilly Media, 2014.
6. Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB, Azat Mardan, Apress, 2015.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Basic RDBMS concepts and SQL

Course Type: Lab

Course Objectives

- To understand the design, querying, storage management and transaction processing of SQL and NoSQL databases.
- To learn advanced database design principles – graph databases, columnar databases and document databases.

Course Outcomes

CO1	Analyze the design of RDBMS and its internals.
CO2	Apply algorithms for query processing and optimization.
CO3	Apply transaction processing and concurrency control techniques for real-world applications.
CO4	Apply the design of Object relational, Graph and NoSQL databases for real-world applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1	1			
CO2	3	2	1	1		
CO3	3	2	1	1	1	
CO4	3	2	2	2	2	1

Syllabus

Overview of RDBMS – Storage and File Structures, Indexing and Hashing - Indexing Structures – Single and Multi-level indexes. Query Processing Optimization and Database Tuning: - Algorithms for Query Processing and Optimization- Physical Database Design and Tuning, Database recovery techniques. Intermediate and Advanced SQL - Embedded SQL Dynamic SQL, Functions and Procedural Constructs, Recursive Queries, Advanced SQL Features.

Transactions Processing and Concurrency Control - Transaction Concept, Transaction model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability. Object Relational Data Models – Complex Data Types, Inheritance, Nesting and Unnesting. NoSQL Databases – NoSQL Data Models, Comparisons of various NoSQL Databases. CAP Theorem, Storage Layout, Query models. Key-Value Stores. Document-databases – Apache CouchDB, MongoDB. Column Oriented Databases – Google’s Big Table, Cassandra - Graph Databases – Cypher basics.

Advanced Application Development – Connecting to MongoDB with Python, MongoDB query Language, Updating/Deleting documents in collection, MongoDB query operators. MongoDB and Python patterns – Using Indexes with MongoDB, GeoSpatial Indexing, Upserts in MongoDB. Document database with Web frameworks.

Text Books/References

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th ed., Pearson, 2017.
2. Database System Concepts, Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, 7th ed., McGraw-Hill, 2019.
3. SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management, Andreas Meier and Michael Kaufmann, Springer, 2019.
4. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, 3rd ed., McGraw-Hill, 2003.
5. Designing Data-Intensive Applications, Martin Kleppmann, O'Reilly Media, 2017.
6. MongoDB: The Definitive Guide, Kristina Chodorow, 3rd ed., O'Reilly Media, 2019.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Basics of Linear Algebra, Probability Theory and Optimization: Vectors, Inner product, Outer product, Inverse of a matrix, Eigen analysis, Probability distributions – Discrete distributions and Continuous distributions; Independence of events, Conditional probability distribution and Joint probability distribution, Bayes theorem, Unconstrained optimization, Constrained optimization.

Course Type: Project

Course Objectives

- To introduce the fundamental concepts and techniques of Machine Learning
- To become familiar with various classification and regression methods
- Apply neural networks, Bayes classifier and k nearest neighbor algorithms in machine learning.
- To develop skills using recent machine learning techniques and solving real world case study

Course Outcomes

CO1	Understand and apply the basics of ML, learning paradigms and concepts of regression.
CO2	Design and develop classifier models and evaluate their performance.
CO3	Acquire skills to build probabilistic models, deep network models for classification and evaluate their performance.
CO4	Develop and build clustering models for real world applications and evaluate their performance.
CO5	Understand and apply the concepts of dimensionality reduction and Machine Learning Model Visualization.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2		1	1	1
CO2	3	3	3	2	2	2
CO3	2	3	2	2	2	1
CO4	3	2	2	2	2	2
CO5	2	2		1	1	1

Syllabus

Introduction to Machine Learning: Overview of machine learning, different forms of learning, data preprocessing, and data visualization; Regression techniques: Linear regression, Ridge regression, Lasso regression, Logistic regression, Discriminant functions and models, Bayesian regression, and regression with basis functions.

Classification and Predictive Modeling: Perceptron, Multilayer Perceptron (MLP), Feedforward Neural Networks, Backpropagation, Support Vector Machines (SVM), Decision Trees, evaluation of classifiers, bias and variance; Gaussian Mixture Models (GMM), Expectation–Maximization (EM) algorithm, Naïve Bayes classifier, Ensemble Methods (Bagging and Boosting), Time Series Prediction, Markov Processes, Introduction to Deep Learning, Convolutional Neural Networks (CNNs), and applications of classification algorithms.

Unsupervised Learning: Clustering techniques: K-means, Hierarchical Clustering, Density-Based Clustering (DBSCAN), assessment of clustering quality; Dimensionality reduction using Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) etc.

Text Books/References

1. Introduction to Machine Learning, Ethem Alpaydin, 4th ed., MIT Press, 2020.
2. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006.
3. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, MIT Press, 2012.
4. Machine Learning, Tom M. Mitchell, McGraw-Hill, 1997.
5. Pattern Classification, Richard O. Duda and Peter E. Hart, 2nd ed., John Wiley & Sons, 2006.
6. Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber, and Jian Pei, 3rd ed., Morgan Kaufmann, 2011.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): None

Course Type: Lab

Course Objectives

- To understand basic principles of Artificial Intelligence
- To understand knowledge representation and reasoning
- To understand the basic areas of artificial intelligence including problem solving, knowledge representation, reasoning, decision making, planning, perception and action
- To understand automatic learning methods in artificial intelligence

Course Outcomes

CO1	Understand formal methods of knowledge representation.
CO2	Understand foundational principles, mathematical tools and program paradigms of AI.
CO3	Apply learning methods for real world scenarios.
CO4	Apply problem solving through search for AI applications.
CO5	Apply logic and reasoning techniques to AI applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	1	2	2	2
CO2	2	2	1	2	2	
CO3	2	2	2	1	1	
CO4	2	2	2	3	3	
CO5	2	1	2	3	3	

Syllabus

Principles of search, uninformed search, informed (heuristic) search, genetic algorithms, game playing - Basic idea behind search algorithms. Complexity. Combinatorial explosion and NP completeness. Polynomial hierarchy. Uninformed Search - Depth-first. Breadth-first. Uniform-cost. Depth-limited. Iterative deepening. Informed search – Best-first. A* search. Heuristics. Hill climbing. Problem of local extrema. Simulated annealing. Genetic Algorithms.

Knowledge bases and inference; constraint satisfaction, logical reasoning - Fuzzy logic. Reasoning under uncertainty – probabilities, conditional independence, Markov blanket, Bayes Nets -

Probabilistic inference, enumeration, variable elimination, approximate inference by stochastic simulation, Markov chain Monte Carlo, Gibbs sampling. Agents that reason logically – Knowledge-based agents. Logic and representation. Propositional (Boolean) logic, Inference in propositional logic. Syntax. Semantics. Probabilistic Reasoning over time: Temporal models, Hidden Markov Models, Kalman filters, Dynamic Bayesian Networks, Automata theory. Planning – Definition and goals. Basic representations for planning. Situation space and plan space.

Inductive learning, concept formation, decision tree learning, statistical approaches, probabilistic methods, learning from examples - neural networks - Probability-Based Learning: Probabilistic Models, Naïve Bayes Models, EM algorithm, Reinforcement Learning.

Text Books/References

1. Artificial Intelligence: A Modern Approach, Stuart J. Russell and Peter Norvig, 3rd ed., Pearson, 2010.
2. A First Course in Artificial Intelligence, Deepak Khemani, McGraw Hill Education (India), 2013.
3. Artificial Intelligence by Example, Denis Rothman, Packt Publishing, 2018.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

ELECTIVES

26CS731

QUANTUM COMPUTING

2-0-2-3

Pre-Requisite(s): Probability

Course Type: Project

Course Objectives

- To understand the components of computing in a Quantum world
- To gain knowledge on mathematical representation of quantum physics and operations.
- To write computations in the real world (standard) in a Quantum computer and simulator.

Course Outcomes

CO1	Understand the fundamental concepts and principles of quantum computing.
CO2	Analyze and implement quantum algorithms using quantum programming frameworks and simulation tools.
CO3	Apply quantum computing techniques to solve computational problems.
CO4	Evaluate the potential of quantum computing for solving real-world problems across diverse application domains.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2		3	2
CO2	2	2	2		3	3
CO3	3	3	3	2	3	3
CO4	3	3	2	2	3	3

Syllabus

Foundations of Quantum Computing: Introduction to quantum computation, postulates of quantum mechanics, Hilbert spaces, qubits, quantum states, superposition, entanglement, measurement, quantum gates, quantum circuits, reversible computation, and the quantum circuit model.

Quantum Algorithms and Programming: Deutsch–Jozsa algorithm, Bernstein–Vazirani algorithm, Simon's algorithm, Quantum Fourier Transform (QFT), Shor's algorithm, Grover's search algorithm and amplitude amplification, quantum programming using Qiskit (or PennyLane), quantum circuit simulation, noisy intermediate-scale quantum (NISQ) computing, and variational quantum algorithms (VQE and QAOA).

Quantum Machine Learning and Applications: Quantum data encoding, variational quantum classifiers, quantum neural networks, Quantum Generative Adversarial Networks (QGANs), quantum annealing, hybrid quantum-classical computing, quantum error correction, quantum communication.

Case studies on applications of quantum computing in cryptography, optimization, healthcare, finance, artificial intelligence, and scientific computing.

Text Books/References

1. Jack D. Hidary, Quantum Computing: An Applied Approach, Springer, 2019.
2. N. David Mermin, Quantum Computer Science: An Introduction, Cambridge University Press, 2007.
3. Michael A. Nielsen and Isaac L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition, Cambridge University Press, 2010.
4. Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press, 2007.
5. Stephen Barnett, Quantum Information, Oxford University Press, 2009.
6. Maria Schuld and Francesco Petruccione, Machine Learning with Quantum Computers, 2nd Edition, Springer, 2021.
7. Buyya, R., & Gill, S. S. (Eds.). Quantum Computing: Principles and Paradigms. Morgan Kaufmann (Elsevier), 2025. ISBN: 9780443290961.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Intermediate knowledge on Object-oriented programming (Java or similar)

Course Type: Project

Course Objectives

- To provide a comprehensive understanding of mobile application development for Android using Kotlin and cross-platform development using Flutter and Dart, with emphasis on application architecture and best practices.
- To develop the skills required to design intuitive user interfaces, manage data and storage, integrate external services and APIs, and build efficient, user-centric mobile applications.

Course Outcomes

CO1	Design and develop interactive mobile user interfaces using Kotlin for Android and Flutter for cross-platform development.
CO2	Implement efficient data management solutions using local storage, cloud databases, and state management techniques.
CO3	Integrate device features, background processing, and external services into mobile applications on both platform platforms.
CO4	Apply modern mobile app architecture principles and best practices to create production-ready applications for Android and Flutter.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	1			
CO2	3	3	2	1		
CO3	3	3	2	2	1	1
CO4	3	3	3	3	2	1

Syllabus

Fundamentals of Mobile App Development: Introduction to mobile application development ecosystems, Android architecture, Flutter framework overview, setting up development environments (Android Studio and Visual Studio Code), Kotlin essentials (syntax, null safety, and functional programming concepts), Dart essentials (syntax and Flutter-specific language features), and creating a basic application structure. Application Components and User Interface Development: Android activities, fragments, and lifecycle; Flutter widgets, state management, and widget lifecycle; user interface design principles; Android XML layouts, ViewGroups, and common UI components; Flutter widgets, layouts, and Material/Cupertino design systems. Navigation and Application Structure: Android intents and Navigation Component; Flutter Navigator, routes, and navigation management.

Data Management and Networking: Local data persistence using Android Shared Preferences and Room Database, Flutter SQLite, networking and REST API integration using Retrofit and Moshi (Android), the http package and JSON serialization (Flutter). Cloud Services and State Management: Firebase integration for Android and Flutter applications, Firebase Authentication, Cloud Firestore, Firebase Cloud Storage, Firebase Cloud Messaging (FCM), state management using ViewModel

and LiveData (Android), and Provider, Riverpod, or BLoC (Flutter). Background Processing and Device Integration: Background processing using WorkManager and Services (Android), Isolates and background execution (Flutter), integration with device features including camera, location, and sensors, and notification services using the Android Notification API, flutter_local_notifications, and Firebase Cloud Messaging.

Advanced User Interface Development: Advanced UI design and animations using MotionLayout and Transitions (Android), Animation Controller and custom animations (Flutter), custom views (Android), and custom widgets and Custom Painter (Flutter). Testing, Performance, and Application Architecture: Unit testing, UI testing, and integration testing using JUnit and Espresso (Android), and Unit, Widget, and Integration Tests (Flutter); performance profiling and optimization; application architecture patterns including MVVM, Repository Pattern, and Clean Architecture; dependency injection using Hilt (Android) and get_it/injectable (Flutter). Application Deployment: Application signing, versioning, packaging, and distribution, and preparing Android and Flutter applications for app store submission.

Text Books/References

1. Alex Forrester, Eran Boudjnah, Alexandru Dumbravan, and Jomar Tigcal, How to Build Android Apps with Kotlin, 2nd Edition, Packt Publishing, 2023. ISBN: 9781837634934.
2. Pierre-Olivier Laurence, Amanda Hinchman-Dominguez, Mike Dunn, and G. Blake Meike, Programming Android with Kotlin, O'Reilly Media, 2021. ISBN: 9781492063001.
3. Peter Späth, Pro Android with Kotlin: Developing Modern Mobile Apps with Kotlin and Jetpack, Apress, 2022. ISBN: 9781484287453.
4. Marco L. Napoli, Flutter Complete Reference: Create Beautiful, Fast and Native Apps for Any Device, Independently Published, 2022.
5. Alberto Miola, Flutter Complete Reference 2.0: The Ultimate Reference for Dart and Flutter, Independently Published, 2023.
6. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, Apress, 2019. ISBN: 9781484251805.
7. Android Developers Documentation – <https://developer.android.com>
8. Flutter Documentation – <https://docs.flutter.dev>
9. Kotlin Documentation – <https://kotlinlang.org/docs>
10. Firebase Documentation – <https://firebase.google.com/docs>

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Software Engineering with Agile and DevOps

Course Type: Project

Course Objectives

- Understand the concepts of software quality, international quality standards, software quality models, and quality assurance processes.
- Develop the ability to design, implement, and evaluate software testing strategies using functional, non-functional, automation, and system testing techniques.
- Apply modern software quality engineering practices, including Agile testing, DevOps, CI/CD, software metrics, defect prevention, and AI-assisted testing to improve software quality.

Course Outcomes

CO1	Understand the principles of software quality, quality models, international standards, and modern software development practices.
CO2	Apply software quality assurance techniques, defect prevention strategies, quality metrics, and risk management practices to improve software quality.
CO3	Design and implement effective software testing strategies using functional, non-functional, automation, and AI-assisted testing techniques.
CO4	Evaluate software systems through verification, validation, system integration, performance, security, usability, and cloud-based testing approaches.
CO5	Develop and manage software quality processes using test planning, CI/CD pipelines, quality engineering practices, and modern testing tools.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	1	1	1
CO2	3	3	1	1	1	1
CO3	3	3	3	3	3	3
CO4	3	3	1	3	3	3
CO5	3	3	1	2	3	2

Syllabus

Software Quality Fundamentals and Quality Assurance: Software quality concepts, quality engineering, quality expectations and customer satisfaction, software quality models (McCall's Quality Model and ISO/IEC 25010), software quality metrics, software process standards (ISO 9001, CMMI, and SPICE (ISO/IEC 330xx)), Agile quality practices, DevOps and continuous quality, software quality assurance framework, verification and validation, defects, errors, faults and failures, defect prevention techniques, root cause analysis, risk-based testing, quality audits, test process improvement, Testing Maturity Model (TMMi), quality monitoring and measurement,

software reliability concepts and reliability models, introduction to Artificial Intelligence (AI) in Software Quality Assurance, and AI-assisted defect prediction.

Software Testing Techniques and Automation: Software Testing Life Cycle (STLC), test planning and test design, test case design techniques (boundary value analysis, equivalence partitioning, decision tables, state transition testing, and pairwise testing), white-box, black-box, grey-box, exploratory, regression, smoke, and sanity testing, API testing, microservices testing, test automation concepts, automation frameworks, Selenium WebDriver, Cypress/Playwright (overview), unit testing frameworks, API testing using Postman, and continuous testing in CI/CD.

System Testing, Non-Functional Testing, and Emerging Trends: System integration testing (top-down, bottom-up, incremental, sandwich, and big bang), acceptance testing, compatibility testing, reliability testing, fault tolerance, hazard analysis, performance testing, load testing, stress testing, security testing, accessibility testing, UI/UX testing, mobile application testing, cloud testing, test reporting and metrics, test management tools (Jira and TestRail), AI-powered test generation, generative AI in software testing, and case studies on web, mobile, and cloud applications.

Text Books/References

1. Kshirasagar Naik and Priyadarshi Tripathy, Software Testing and Quality Assurance: Theory and Practice, 2nd Edition, Wiley India, 2011.
2. Paul C. Jorgensen, Software Testing: A Craftsman's Approach, 5th Edition, CRC Press, 2024.
3. Jeff Tian, Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Wiley, 2005.
4. Lisa Crispin and Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley Professional, 2009.
5. Mark Winteringham, Software Testing with Generative AI, Manning Publications, 2024.
6. Rex Black, Managing the Testing Process, 4th Edition, Wiley, 2023.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Distributed Systems, Basic OS, Networks knowledge

Course Type: Project

Course Objectives

- Discusses major components of Fog and Edge computing architecture such as middleware, interaction protocols, and autonomic management
- Perform data collection, learning and analytics at the edge
- Improve performance at the edge and analyze the latest edge-based systems, platforms, and design applications

Course Outcomes

CO1	Understand the fundamentals of Fog and Edge Computing, including their architecture, enabling technologies, and role in IoT ecosystems.
CO2	Design Fog and Edge Computing systems using reference architectures, middleware, and resource management techniques.
CO3	Apply data management, edge analytics, and intelligent computing techniques to develop Fog and Edge Computing applications.
CO4	Analyze optimization, security, and deployment strategies for efficient Fog and Edge Computing solutions.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	2			
CO2	2	3	2	2		
CO3	2	2	2			
CO4	2	2	2			

Syllabus

Foundations: Evolution from Cloud Computing to Fog and Edge Computing; Internet of Things (IoT) and the need for Edge Computing; Hierarchy and characteristics of Cloud, Fog, and Edge Computing; Edge networks and communication paradigms; Edge Computing architectures; OpenFog Reference Architecture; Optimization techniques in Fog and Edge environments; Case studies on Smart Cities and Industrial IoT.

Middleware, Data Management, and Security: Edge middleware and resource management; Virtualization and containerization using Docker; Data management and storage at the Edge; Edge analytics; Introduction to Edge AI and TinyML; Task offloading and resource scheduling in Edge environments; Security, privacy, and trust challenges in IoT-enabled Fog and Edge Computing.

Applications: Applications of Edge Computing in Smart Healthcare, Smart Transportation, Industry 4.0 and Digital Twins, Smart Agriculture, Environmental Monitoring, Smart Surveillance,

UAV/Drone-enabled Edge Computing, and 5G-enabled Edge Computing; Modeling, simulation, and performance evaluation using the iFogSim2 toolkit; Design, implementation, and deployment of an Edge Computing application.

Text Books/References

1. Buyya, R., & Srirama, S. N. (Eds.). Fog and Edge Computing: Principles and Paradigms. John Wiley & Sons, 2019. ISBN: 9781119524984.
2. Javid Taheri and Shuiguang Deng (Eds.), Edge Computing: Models, Technologies and Applications, Institution of Engineering and Technology (IET), 2020.
3. Khaldoun Al Agha, Pauline Loygue, and Guy Pujolle, Edge Networking, Wiley-ISTE, 2022.
4. Xin Sun and Amin Vahdat, Edge Computing: A Primer, CRC Press, 2019.
5. Industry IoT Consortium (IIC), OpenFog Reference Architecture for Fog Computing, 2017.
6. IEEE Standard 1934-2018, IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing, IEEE, 2018.
7. iFogSim2 Documentation and User Guide (Official Documentation).

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): C/C++ Programming and Data Structure

Course Type: Project

Course Objectives

- The student gains a fundamental understanding of virtual reality (VR), augmented reality (AR), and mixed reality (MR) technologies. This includes knowledge of the devices used, the process of building virtual environments, and the various methods of interaction and modeling.
- Explore the key applications of virtual reality (VR) and augmented reality (AR) technologies across sectors such as education, healthcare, industry, and entertainment.
- This course enables students to explore research topics in augmented reality (AR) and virtual reality (VR).

Course Outcomes

CO1	Understand the fundamental concepts, technologies, and design principles of Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR).
CO2	Understand the mathematical foundations, hardware, software development tools, and interaction techniques used in AR/VR systems.
CO3	Design and develop immersive AR/VR applications by applying principles of user interface design, interaction design, and user experience using appropriate hardware, software frameworks and development platforms.
CO4	Analyze human factors, user performance, sensorial conflicts, and emerging technologies in AR/VR/MR systems.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2	1	1	
CO2	3	2	2	1	1	
CO3	3	2	3	3	3	1
CO4	3	2	2	1	1	

Syllabus

Foundations of Virtual, Augmented, and Mixed Reality: Introduction to Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR); taxonomy, characteristics, and applications of VR, AR, and MR; differences between VR, AR, and MR; challenges in AR/VR systems; geometry of virtual worlds; geometric transformations, position and orientation, viewing transformations; fundamentals of the human visual system, depth cues, and stereopsis.

VR and AR Systems, Hardware, and Design: VR systems, input and output devices, VR hardware (head-mounted displays, controllers, and gloves), AR systems and types (marker-based, markerless, and location-based), AR hardware (smart glasses, wearable devices, and smartphones), software development tools for VR and AR, user interface (UI) and user experience (UX) design principles,

spatial design, interaction design, visual design, feedback, and responsiveness.

Advanced Topics and Applications in AR/VR/MR: Mixed Reality (MR), Artificial Intelligence for AR/VR, computer vision in AR/VR, IoT-enabled AR/VR systems, human factors in AR/VR/MR, user performance, sensorial conflicts, emerging trends and innovations, and case studies on applications of AR/VR/MR in healthcare, education, manufacturing, gaming, entertainment, retail, and smart cities.

Text Books/References

1. Grigore C. Burdea and Philippe Coiffet, Virtual Reality Technology, 2nd Edition, Wiley-IEEE Press, 2003.
2. Alan B. Craig, Understanding Augmented Reality: Concepts and Applications, Morgan Kaufmann, 2013.
3. Alan B. Craig, William R. Sherman, and Jeffrey D. Will, Developing Virtual Reality Applications: Foundations of Effective Design, Morgan Kaufmann, 2009.
4. Vladimir Geroimenko (Ed.), Augmented Reality and Artificial Intelligence, Springer Nature Switzerland, 2024.
5. Gitanjali Rahul Shinde, Prashant Shantaram Dhotre, Parikshit Narendra Mahalle, and Nilanjan Dey, Internet of Things Integrated Augmented Reality, Springer Nature Singapore, 2020.
6. Dieter Schmalstieg and Tobias Höllerer, Augmented Reality: Principles and Practice, Addison-Wesley Professional, 2016.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Basics of Computer Science

Course Type: Project

Course Objectives

- Understand how blockchain systems (mainly Bitcoin and Ethereum) work
- To securely interact with them
- Design, build, and deploy smart contracts and distributed applications
- Integrate ideas from blockchain technology into their own projects

Course Outcomes

CO1	Understand the concepts of cryptocurrency, blockchain, and distributed ledger technologies.
CO2	Analyze blockchain architectures, consensus mechanisms, and their applications across different domains.
CO3	Evaluate security, privacy, and scalability challenges associated with blockchain and cryptocurrency systems.
CO4	Design and assess blockchain-based solutions for real-world applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2	2		
CO2	2	2	2	2		
CO3	2	3	2	3		2
CO4	3	3	3	3	2	

Syllabus

Fundamentals of Blockchain and Bitcoin: Introduction to Blockchain: History, evolution, characteristics, types, applications, and benefits; Financial systems and the need for Blockchain; Blockchain architecture and Distributed Ledger Technology (DLT); Decentralization and distributed consensus; Consensus mechanisms; CAP theorem in Blockchain; Cryptographic foundations: hash functions, symmetric and asymmetric cryptography, digital signatures, public and private keys; Overview of Blockchain platforms. Bitcoin: Blockchain structure, transactions, wallets, payments, mining, and Bitcoin ecosystem.

Smart Contracts and Ethereum: Smart Contracts: Concepts, characteristics, lifecycle, advantages, applications, Ricardian contracts, and Blockchain Oracles. Ethereum: Architecture and design principles; Ethereum accounts, Ether, Gas, Ethereum Virtual Machine (EVM), transactions, mining, and consensus mechanisms; Development of Ethereum smart contracts using Solidity; Decentralized Applications (DApps) and their architecture.

Enterprise Blockchain and Applications: Enterprise Blockchain: Concepts and requirements; Hyperledger architecture and ecosystem; Hyperledger Fabric components, peers, ordering service, channels, Membership Service Provider (MSP), chaincode, and transaction flow; Comparative

analysis of Ethereum, Hyperledger Fabric, and other Blockchain platforms; Case studies and applications in finance, healthcare, supply chain management, digital identity, IoT, and other enterprise domains.

Text Books/References

1. Imran Bashir, Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, 4th Edition, Packt Publishing, 2023. (If the latest edition is unavailable, the 2nd Edition, 2018, is acceptable.)
2. Andreas M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, 2nd Edition, O'Reilly Media, 2017.
3. Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress, 2017.
4. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
5. William Mougayar, The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, Wiley, 2016.
6. Nitin Gaur, Luc Desrosiers, Petr Novotny, Venkatraman Ramakrishna, and Eric Ramanathan, Hands-On Blockchain with Hyperledger, Packt Publishing, 2019.
7. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
8. Vitalik Buterin, Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform, 2014.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Computer Networks, Operating Systems, Programming.

Course Type: Project

Course Objectives

- This course introduces the basic principles of cloud computing, cloud native application development and deployment, containerization principles, micro-services and application scaling.
- To learn Virtualization and its types in cloud computing.
- It will also equip the students to understand major industry players in the public cloud domain for application development and deployment.

Course Outcomes

CO1	Understand the fundamental concepts, architecture, and enabling technologies of Cloud Computing.
CO2	Analyze cloud service models, deployment models, and cloud infrastructure for diverse application scenarios.
CO3	Apply virtualization, containerization, and cloud-native technologies for efficient application deployment.
CO4	Design and deploy cloud-native applications using containerization and container orchestration platforms.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1			1	1
CO2	1	2	3	3	1	1
CO3		1	1	1	2	2
CO4	1	1	1	2	2	2

Syllabus

Foundations of Cloud Computing: Introduction to Cloud Computing; distributed computing taxonomy (Cluster, Grid, Peer-to-Peer, Utility, Cloud, Edge, and Fog Computing); characteristics, benefits, challenges, and use cases of cloud computing; cloud service models (XaaS); cloud deployment models (public, private, hybrid, and community); major public cloud providers; cloud security challenges; and an overview of cognitive cloud services.

Cloud Services and Application Deployment: Cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS); compute, storage, networking, identity, middleware, and database services; cloud-native application development; microservices architecture; web application deployment using public cloud platforms; cloud resource management, monitoring, and automation; and case studies on architecting cloud-based solutions.

Virtualization, Containers, and Cloud Orchestration: Virtualization fundamentals; hypervisors and virtualization techniques; containerization concepts; Docker architecture, container images, registries, Docker Compose, and container deployment; container orchestration concepts; Kubernetes architecture, components, YAML-based deployment, Minikube, service discovery, scaling, rolling updates, and an overview of alternative orchestration platforms; cloud application deployment and best practices.

Text Books/References

1. Buyya, R., Vecchiola, C., Selvi, S. T., Poojara, S., & Srirama, S. N. Mastering Cloud Computing: Foundations and Applications Programming. 2nd Edition. Morgan Kaufmann (Elsevier), 2026. ISBN: 978-0-443-40435-1
2. Buyya, R., Vecchiola, C., Selvi, T., Poojara, S., & Srirama, S. N. Mastering Cloud Computing: Powering AI, Big Data, and IoT Applications. 2nd Edition. McGraw Hill Education (India), 2024.
3. Arshdeep Bahga and Vijay Madisetti, Cloud Computing: A Hands-On Approach, Universities Press, 2014.
4. Karl Matthias and Sean P. Kane, Docker: Up & Running: Shipping Reliable Containers in Production, 3rd Edition, O'Reilly Media, 2021.
5. Eberhard Wolff, Microservices: Flexible Software Architecture, Addison-Wesley Professional, 2016.
6. Aurobindo Sarkar and Amit Shah, Learning AWS: Design, Build, and Deploy Responsive Applications Using AWS Cloud Components, Packt Publishing, 2018.
7. Nigel Poulton, The Kubernetes Book, 2024 Edition.
8. Kelsey Hightower, Brendan Burns, and Joe Beda, Kubernetes: Up & Running, 3rd Edition, O'Reilly Media, 2022.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): None

Course Type: Project

Course Objectives

- To introduce the basics of Neural networks and Deep neural networks.
- To explore the major deep network architectures including convolutional and recurrent.
- To enable and apply different deep neural network models.
- To apply various deep learning techniques to provide solutions for real world problems.

Course Outcomes

CO1	Understand the foundational concepts behind Neural Networks, Deep Learning and Deep Neural Network architectures.
CO2	Apply deep learning frameworks and techniques to develop and train deep neural network models.
CO3	Analyze the performance of custom-built and pre-trained deep learning models for various applications.
CO4	Design and implement appropriate deep learning solutions for real-world problems using modern architectures and frameworks.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1	2	2		
CO2	3	2	1	2	2	1
CO3	3	2	2	1	2	1
CO4	3	2	1	1	1	1

Syllabus

Foundations of Neural Networks and Deep Learning: Introduction to Artificial Neural Networks and Deep Learning. Basics of image and signal processing with convolution operation. Perceptron, Multi-Layer Perceptron (MLP), Feedforward Neural Networks, Backpropagation algorithm, training, validation and testing of neural networks. Activation functions, loss functions, hyperparameter tuning, optimization algorithms including Gradient Descent, Stochastic Gradient Descent (SGD), and Adam, regularization techniques including L1/L2 regularization, dropout, early stopping, and Batch Normalization.

Deep Neural Network Architectures: Common architectural principles and building blocks of deep neural networks. Convolutional Neural Networks (CNNs) and Deep CNN architectures. Recurrent Neural Networks (RNNs), vanishing gradient problem, Long Short-Term Memory (LSTM) units, and sequence modelling. Transfer Learning and fine-tuning of pre-trained models. Introduction to Generative AI, Foundation Models, and Transformer-based architecture.

Representation Learning and Advanced Applications: Representation Learning and Autoencoders, including Sparse and Denoising Autoencoders. Applications of deep learning in computer vision and natural language processing, including Named Entity Recognition, Sentiment Analysis, Opinion Mining, Question Answering, and Neural Text Summarization. Overview of deep learning frameworks (TensorFlow/PyTorch). Introduction to Large Language Models (LLMs), prompt engineering, and recent advances in Deep Learning.

Text Books/References

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 1st Edition, 2016.
2. François Chollet, Deep Learning with Python, Manning Publications, 2nd Edition, 2021.
3. Josh Patterson and Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Media, 1st Edition, 2017.
4. Nikhil Buduma and Nicholas Locascio, Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, O'Reilly Media, 2nd Edition, 2021.
5. Nikhil Ketkar, Deep Learning with Python: A Hands-on Introduction, Apress, 1st Edition, 2017.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester Exam/Project		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination or as a project presentation and evaluation, as appropriate to the nature of the course and subject to the approval of the Department.

Pre-Requisite(s): Advanced Networks

Course Type: Lab/Project

Course Objectives

- Good hands-on exposure to various IoT enabling technologies
- Good practical knowledge in various application layer protocols for IoT systems.
- Practical exposure in design and deployment of IoT systems for specific use case.

Course Outcomes

CO1	Understand the fundamental concepts of Internet of Things (IoT), IoT architecture, and enabling technologies.
CO2	Develop IoT applications using appropriate hardware platforms, sensors, and software frameworks.
CO3	Implement and evaluate IoT communication protocols across different layers of IoT architecture.
CO4	Design and deploy IoT-based solutions for real-world applications and domain-specific use cases.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2			
CO2	2	3	2	2	2	2
CO3	2	2	2			
CO4	2	3	2	2	2	2

Syllabus

Introduction to the Internet of Things (IoT): Definition, characteristics, IoT ecosystem, complete IoT architectural stack, enabling technologies, and challenges. Sensors and hardware for IoT, hardware platforms including Arduino, Raspberry Pi, and microcontrollers with wireless interfaces. Data acquisition from sensors and a case study using a representative IoT hardware platform.

IoT communication protocols: Application layer protocols (MQTT, CoAP, WebSockets), Internet and network layer protocols (IPv4/IPv6, RPL), identification using URIs, transport technologies including Wi-Fi, Li-Fi, and Bluetooth Low Energy (BLE). Device discovery, service discovery and management, data protocols, and device management. IoT-specific real-time databases, user interface design, IoT analytics, privacy, security, and vulnerability management. Machine Learning techniques for securing IoT systems. Integration of IoT end devices with Edge and Cloud computing environments.

IoT application architectures and design considerations. Architectural analysis and case studies of IoT applications in Smart Cities, Smart Water Management, Smart Agriculture, Smart Energy, Smart Healthcare, Smart Transportation, Smart Retail, Smart Waste Management, and Smart

Manufacturing. Emerging trends in intelligent IoT systems and AI-enabled IoT applications.

Text Books/References

1. Pethuru Raj and Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press, 1st Edition, 2017.
2. Sudip Misra, Anandarup Mukherjee, and Arijit Roy, Introduction to IoT, Cambridge University Press, 1st Edition, 2022.
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, and Jerome Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, 1st Edition, 2017.
4. Rodrigo J. Hernández, Practical IoT Handbook: Programming IoT by Implementing Hands-on Projects with Arduino, Python, and Raspberry Pi, BPB Publications, 1st Edition, 2025.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Computer Networks

Course Type: Lab

Course Objectives

- Understand the basic operation and design of a wireless system.
- Analyze the performance issues of modern and advanced wireless networks.
- Understand the dynamics of wireless environment and means of communication across heterogeneous networks

Course Outcomes

CO1	Understand the fundamental principles, architectures, and protocols of existing and emerging wireless networking technologies.
CO2	Analyze the design principles, standards, and applications of modern Wireless Local Area Networks (WLANs), including Wi-Fi technologies.
CO3	Analyze the architecture, standards, and applications of modern Wireless Personal Area Networks (WPANs), including Bluetooth and related technologies.
CO4	Understand the fundamentals of 4G and 5G cellular networks and design suitable applications based on emerging cellular communication technologies.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			
CO2	2	2	2	2		
CO3	2	2	2	2		
CO4	2	2	2	2		

Syllabus

Fundamentals of Wireless Communication and Mobile Networks: Introduction to wireless communication systems and mobile networks. Basic wireless propagation mechanisms: reflection, diffraction, and scattering. Free space path loss model and wireless channel characteristics. Basic terminologies including receiver sensitivity, Signal-to-Noise Ratio (SNR), and Received Signal Strength (RSS). Introduction to multipath propagation, Line-of-Sight (LoS), Non-Line-of-Sight (NLoS) communication, and Inter-Symbol Interference (ISI). Overview of modern wireless and mobile communication systems: Wireless Local Area Networks (WLAN), Wireless Personal Area Networks (WPAN), and Cellular Networks. Principles of cellular networks and evolution of mobile communication technologies from 1G, 2G, 3G, 4G, and 5G. Overview of emerging 6G technologies and future wireless communication trends.

Wireless Local and Personal Area Networks: Modern Wireless Local Area Networks: Wi-Fi fundamentals, IEEE 802.11 architecture, ISM frequency bands, IEEE 802.11 channels, physical layer concepts, hidden node problem, collision avoidance mechanisms, and IEEE 802.11 MAC layer

protocol using CSMA/CA. IEEE 802.11 architecture, frame format, power management, and enhanced Quality of Service using IEEE 802.11e. Recent WLAN standards including IEEE 802.11n, IEEE 802.11ac, and IEEE 802.11ax. Modern Wireless Personal Area Networks: Introduction to Bluetooth technology and versions, Bluetooth 1.1, Bluetooth 4.0 with Bluetooth Low Energy (BLE), Bluetooth 5.0, and Bluetooth 5.3.

Cellular Networks and Emerging Wireless Technologies: Multiple access techniques including Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA). Modern cellular communication systems: 4G and 5G wireless systems and standards, LTE network architecture and protocols, LTE Advanced, 5G network architecture, and an overview of emerging 6G communication technologies. Hands-on experience with Wi-Fi/Bluetooth packet capturing and analysis using packet capture tools.

Text Books/References

1. Hassan, M., Wireless and Mobile Networking, 1st Edition, CRC Press, 2022. DOI: 10.1201/9781003042600.
2. Agha, K. A., Pujolle, G., and Yahiya, T. A., Mobile and Wireless Networks, 1st Edition, Wiley, 2016.
3. Theodore S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 2nd Edition.
4. William Stallings, Wireless Communications & Networks, Pearson Education India, 2009.
5. Matthew Gast, 802.11n: A Survival Guide, O'Reilly Media.
6. Matthew Gast, 802.11ac: A Survival Guide, O'Reilly Media.
7. Pei Zheng, Larry L. Peterson, Bruce S. Davie, and Adrian Farrel, Wireless Networking Complete, Morgan Kaufmann, 2009
8. Stallings, William, and Beard, Cory, Wireless Communication Networks and Systems, Pearson, 2015.
9. Morais, D. H., 5G NR, Wi-Fi 6, and Bluetooth LE 5: A Primer on Smartphone Wireless Technologies, Springer Nature, 2023.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Basic knowledge of computer networks, storage, and operating systems

Course Type: Lab

Course Objectives

- To understand the core principles of software-defined systems.
- To explore the architecture and components of software-defined networking, storage, and compute.
- To learn about the benefits and challenges of implementing software-defined systems.
- To gain hands-on experience with tools and platforms used in software-defined systems.

Course Outcomes

CO1	Understand the core concepts, principles, and terminologies associated with Software-Defined Systems (SDS).
CO2	Analyze the design principles, algorithms, and software architectures used in developing software-defined systems.
CO3	Evaluate orchestration and management strategies for software-defined systems, including automation techniques and DevOps practices.
CO4	Develop and configure software-defined systems using appropriate tools, frameworks, and platforms for real-world applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	3	3	1	1	1
CO2	3	3	1	1	1	1
CO3	3	3	3	3	3	3
CO4	3	3	1	3	3	3

Syllabus

Introduction to Software Defined (SD) Computing: evolution, history, key concepts, terminologies, and software-defined infrastructures. Overview of Software Defined Networking (SDN), Software Defined Storage (SDS), and Software Defined Data Centers (SDDC). Principles of SDN, SDN architecture, components, control plane and data plane separation, OpenFlow protocol, and other SDN protocols. SDN applications and case studies.

Network Function Virtualization (NFV): architecture, components, and Virtual Network Functions (VNFs). NFV use cases and case studies. Software Defined Storage: fundamentals, architectures, models, implementation strategies, and applications. Software Defined Data Centers (SDDC): concepts, components, enabling technologies, implementation challenges, and real-world applications. Software-defined networking in cloud data centers and hybrid/multi-cloud environments.

Kubernetes-based orchestration of software-defined infrastructure. Cloud-Native Network Functions (CNFs), containerized infrastructure management, and orchestration strategies. Software-defined security, Zero Trust architecture, and security considerations in SDS. AI-assisted infrastructure management (AIOps). Software Defined Systems management: orchestration, monitoring, performance optimization, and resource management. Automation and DevOps practices, CI/CD pipelines, infrastructure-as-code concepts, and emerging trends in software-defined systems.

Tools and Frameworks: Red Hat OpenStack Platform (RHOSP), Mininet, Open vSwitch, OpenDaylight/ONOS, Docker, and Kubernetes.

Integrated Laboratory / Project: Hands-on implementation of mini SDN environments using Mininet and SDN controllers, implementation of NFV scenarios using virtual machines or containers, automation of infrastructure deployment using scripting and orchestration tools, Kubernetes-managed software-defined application deployment, software-defined security policy implementation and evaluation, performance monitoring and visualization of software-defined systems, and comparative study of OpenStack and Kubernetes-based software-defined infrastructures.

Text Books/References

1. Oswald Coker and Siamak Azodolmolky, Software Defined Networking with OpenFlow, 2nd Edition, O'Reilly Media, 2017.
2. Paul Goransson, Chuck Black, and Timothy Culver, Software Defined Networks: A Comprehensive Approach, 2nd Edition, Morgan Kaufmann, 2016.
3. Ken Gray and Thomas D. Nadeau, Network Function Virtualization: Concepts and Applications, Morgan Kaufmann, 2016.
4. Deze Zeng, Lin Gu, Shengli Pan, and Song Guo, Software Defined Systems: Sensing, Communication and Computation, SpringerBriefs in Computer Science, Springer, 2020.
5. Nigel Poulton, *The Kubernetes Book: The Fastest Way to Get Your Head Around Kubernetes*, Leanpub, Latest Edition.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): None

Course Type: Lab/Project

Course Objectives

- To understand the Big Data technologies and tools for storing and processing
- To develop skills in analyzing and visualizing large datasets.
- To explore machine learning for predictive analytics in Big Data Analytics

Course Outcomes

CO1	Understand the fundamental concepts, characteristics, and significance of Big Data in modern data-driven environments.
CO2	Apply Big Data technologies, frameworks, and tools for efficient storage, retrieval, and processing of large-scale datasets.
CO3	Analyze and visualize Big Data to extract meaningful insights and support data-driven decision-making.
CO4	Design and develop machine learning models and analytics solutions using Big Data frameworks for predictive analysis.
CO5	Apply functional programming concepts and distributed computing principles for optimizing Big Data processing and management.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1	1			1
CO2	3	2	1		2	2
CO3	2	2	1	1	2	2
CO4	2	2	1	1	2	1
CO5	1	1	1	1	2	1

Syllabus

Introduction to Big Data and Hadoop: Introduction, characteristics, sources, and types of Big Data; Challenges in Big Data processing; Distributed computing concepts; NoSQL databases and data models; Hadoop ecosystem: Hadoop Distributed File System (HDFS), Hadoop YARN, MapReduce framework, and Hadoop ecosystem components; MapReduce programming model: Mapper, Reducer, data flow, and real-world applications.

Apache Spark, Hive, and Pig: Spark architecture, execution model, Resilient Distributed Datasets (RDDs), transformations and actions, DataFrames, Spark SQL, and data preprocessing; Apache Hive: Architecture, Hive data types, file formats, Hive Query Language (HQL), and User Defined Functions (UDFs); Apache Pig: Architecture, Pig Latin language, data processing, and analytics using Pig.

Machine Learning and Analytics using Spark: Spark MLlib, Machine learning pipelines, feature engineering, model building, and evaluation; Regression, Classification, Clustering, and Recommendation techniques using Spark; Spark Structured Streaming for real-time analytics; Performance optimization in Spark applications; Applications and case studies of Big Data Analytics in healthcare, finance, social media, smart cities, and Internet of Things (IoT).

Text Books/References

1. Seema Acharya and Subhashini Chellappan, Big Data and Analytics, Wiley Publications, 1st Edition, 2015.
2. Tom White, Hadoop: The Definitive Guide, O'Reilly Media, 3rd Edition, 2011.
3. Karau, Holden, Polak, Adi, and Warren, Rachel, High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark, O'Reilly Media, Inc., 2026.
4. Drabas, T., and Lee, D., Learning PySpark, Packt Publishing Ltd., 2017.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Computer Organization and Architecture or equivalent.

Course Type: Lab

Course Objectives

- To comprehend Parallel Processing Fundamentals and Programming
- To develop Proficiency in HPC Programming Tools
- To explore modern HPC and GPU based heterogeneous architectures and technologies for real world applications.
- To evaluate and innovate in HPC Design through cloud environments

Course Outcomes

CO1	Understand and apply various levels of parallelism including instruction, transaction, task, thread, memory, function, and data flow models.
CO2	Develop, implement, and optimize parallel programs using modern parallel programming models and tools such as OpenMP, MPI, Pthreads, Cilk, and CUDA.
CO3	Analyze and evaluate scalable storage architectures, including RAID, SSD cache, SAS, and SAN, for high-performance computing systems.
CO4	Evaluate the capabilities, limitations, and applications of emerging HPC technologies, including accelerated computing, cloud-based HPC, and next-generation computing systems.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	1	2	2	2	
CO2	2	2	1			
CO3	2	2	2	2	2	
CO4	1	1	2	2	2	

Syllabus

Introduction to High Performance Computing (HPC): concepts, applications, and evolution. HPC architectures: shared memory and distributed memory architectures, multiprocessor architectures, and parallel processing concepts. Levels and models of parallelism: instruction-level, transaction-level, task-level, thread-level, memory-level, function-level, data flow models, and demand-driven computation. Parallel architecture: superscalar architecture, multi-core processors, multi-threaded systems, servers, and cloud-based HPC environments. Fundamental design issues in HPC: load balancing, scheduling, synchronization, and resource management. Limitations in HPC: bandwidth, latency, and latency hiding techniques. HPC benchmarking: scientific, engineering, commercial applications, and workload characterization.

Parallel programming languages and environments: OpenMP, POSIX Threads (Pthreads), Message Passing Interface (MPI), Java parallel programming, and Cilk. Performance analysis of parallel algorithms, scalability, speedup, efficiency, and performance evaluation techniques. Scalable

storage systems: RAID, SSD cache, SAS, and Storage Area Networks (SAN). HPC systems based on cluster, cloud, and grid computing: architectures, economic models, infrastructure-as-a-service, platform-as-a-service, and computation-as-a-service. Case studies on parallel programming using OpenMP, Pthreads, MPI, and Cilk.

Accelerated HPC architectures: GPU, FPGA, Intel Xeon Phi, and Cell Broadband Engine (Cell BE) based computing systems. GPU architecture and evolution, General Purpose GPU (GPGPU) computing, CUDA programming model, thread execution, memory hierarchy, and parallel execution in CUDA. Power-aware HPC design: computing and communication optimization, processor and memory design, interconnect design, and power management techniques. Advanced topics in HPC: petascale computing, Big Data processing using HPC, optical interconnects in HPC, and quantum computing. Case studies on HPC clusters, GPU-accelerated systems, and emerging HPC applications.

Text Books/References

1. George Hager and Gerhard Wellein, Introduction to High-Performance Computing for Scientists and Engineers, CRC Press, Taylor & Francis Group, Revised Edition, 2019.
2. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, 1st Edition, Addison-Wesley Professional, 2010.
3. Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis, Introduction to Parallel Computing, 2nd Edition, Pearson Education, 2003.
4. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, 5th Edition, Elsevier, 2011.
5. David B. Kirk and Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 1st Edition, Elsevier, 2010.
6. Michael T. Heath, Scientific Computing: An Introductory Survey, 2nd Edition, McGraw Hill Education, 2011.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Basic knowledge of Deep Learning, Familiarity with Python programming

Course Type: Lab

Course Objectives

- To understand the principles and architectures of modern generative AI models.
- To study GANs, diffusion models, and LLMs for image and text generation.
- To implement generative AI models and LLM-based applications.
- To apply prompting, fine-tuning, and RAG for language-based applications.
- To evaluate generative AI models using suitable performance measures.

Course Outcomes

CO1	Understand the fundamentals of generative models and their major architecture.
CO2	Design and implement generative models for image generation and related applications.
CO3	Apply transformers and LLM techniques for language generation and NLP tasks.
CO4	Develop LLM-based applications using RAG and introductory agentic AI workflows.
CO5	Evaluate and compare generative AI models using suitable performance measures.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1				
CO2	2	3	2			
CO3	3	2	2			
CO4	3	1		2	1	1
CO5	3	1		2		

Syllabus

Recap of Deep Neural Networks, Autoencoders, Variational Autoencoders (VAEs), attention mechanisms, and fundamentals of generative modelling. Introduction to Generative Adversarial Networks (GANs): generator and discriminator architectures, adversarial training, loss functions, and optimization challenges. Advanced GAN architectures: Deep Convolutional GANs (DCGAN), Conditional GANs, Pix2Pix, CycleGAN, and StyleGAN. Applications of GANs in image synthesis, data augmentation, and content generation. Challenges in GANs including mode collapse, bias, fairness, and privacy preservation.

Introduction to diffusion models: principles of forward noising and reverse denoising processes. Denoising Diffusion Probabilistic Models (DDPMs), score-based generative models, latent diffusion models, and Stable Diffusion. Applications of diffusion models in text-to-image generation, image editing, inpainting, and super-resolution. Comparative analysis of VAEs, GANs, and diffusion models. Evaluation techniques for generated images including Fréchet Inception Distance (FID), diversity, realism assessment, and human evaluation methods.

Principles of Transformer architecture: self-attention mechanisms, positional encoding, tokenization, embeddings, and transformer variants including encoder-only, decoder-only, and encoder-decoder

models. Overview of BERT and GPT-based models, pretraining and fine-tuning strategies. Prompt engineering, instruction tuning, supervised fine-tuning, parameter-efficient fine-tuning techniques including LoRA and QLoRA, quantization, small language models, Reinforcement Learning from Human Feedback (RLHF), Direct Preference Optimization (DPO), and LLM alignment. Applications of Large Language Models in text generation, summarization, translation, question answering, sentiment analysis, chatbots, healthcare, and code generation. Retrieval-Augmented Generation (RAG), embeddings, vector databases, semantic search, LLM evaluation, and Introduction to Agentic AI: LLM agents, tool usage, memory, planning, workflow orchestration, and multi-agent systems.

Text Books/References

1. Bernad Marr, Generative AI in Practice: 100+ Amazing Ways Generative Artificial Intelligence is Changing Business and Society, Wiley, 2024.
2. Jay Alamar and Maarten Grootendorst, Hands-On Large Language Models, O'Reilly Media, 2024.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
4. Jason Brownlee, Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems, Machine Learning Mastery, 2017.
5. Sudharsan Ravichandiran, Getting Started with Google BERT: Build and Train State-of-the-Art Natural Language Processing Models Using BERT, Packt Publishing, 2021.
6. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition (Draft), available online, Stanford University.
7. Jay Alamar and Maarten Grootendorst, An Illustrated Guide to AI Agents, O'Reilly Media.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): None

Course Type: Lab

Course Objectives

- To introduce the ethical principles, challenges, and responsible practices associated with the design and deployment of Artificial Intelligence systems.
- To develop an understanding of fairness, privacy, security, transparency, and governance issues in AI-driven applications.
- To enable students to analyze and design responsible AI solutions considering societal, legal, and ethical implications.

Course Outcomes

CO1	Understand the fundamental concepts of AI ethics, moral philosophy, human values, and principles of responsible and trustworthy AI.
CO2	Apply ethical reasoning and responsible AI principles to address challenges in the design, development, and deployment of AI systems.
CO3	Evaluate AI applications based on privacy, fairness, transparency, accountability, security, and governance principles.
CO4	Analyze the ethical, societal, legal, and environmental impacts of AI technologies through real-world case studies and emerging AI applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1	2	2	2	2	1
CO2	2	3	2	2	2	1
CO3	2	3	2	3	2	1
CO4	2	2	3	2	3	2

Syllabus

Introduction to AI Ethics: history and evolution of AI ethics, ethical theories and moral philosophy, human values in AI, responsible AI principles, and trustworthy AI. Fairness, Accountability, Transparency, and Ethics (FATE) framework. Algorithmic bias and discrimination: sources of bias in data and models, fairness metrics, bias detection, and mitigation techniques. Human-centered AI, ethical decision-making in intelligent systems, responsible AI practices, and case studies.

Privacy and data ethics: data protection principles, informed consent, data ownership, and data governance. Privacy-preserving machine learning techniques including federated learning and differential privacy. AI security and robustness: adversarial attacks, defense mechanisms, and secure AI development. Explainable Artificial Intelligence (XAI): interpretability, transparency, and accountability in AI systems. Human oversight, ethical AI development lifecycle, and responsible AI tools and frameworks.

AI governance and regulation: global AI governance frameworks, OECD AI Principles, UNESCO Recommendation on the Ethics of Artificial Intelligence, European Union AI Act, and comparative analysis of AI governance approaches. AI governance in India: Responsible AI for All, IndiaAI initiatives, and AI applications in public services. Ethical challenges in Generative AI and Foundation Models, intellectual property issues, societal impact of AI, environmental sustainability, AI for social good, sector-specific case studies, and emerging challenges in responsible AI.

Text Books/References

1. Mark Coeckelbergh, AI Ethics, MIT Press, 2020.
2. Virginia Dignum, Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way, Springer, 2019.
3. Michael Kearns and Aaron Roth, The Ethical Algorithm: The Science of Socially Aware Algorithm Design, Oxford University Press, 2019.
4. Wendell Wallach and Colin Allen, Moral Machines: Teaching Robots Right from Wrong, Oxford University Press, 2008.
5. NITI Aayog, National Strategy for Artificial Intelligence (#AIForAll), Government of India.
6. IndiaAI Mission, Ministry of Electronics and Information Technology (MeitY), Government of India.
7. Ministry of Electronics and Information Technology, Digital Personal Data Protection Act, 2023.
8. UNESCO, Recommendation on the Ethics of Artificial Intelligence, 2021.
9. OECD, OECD AI Principles.
10. NIST, AI Risk Management Framework (AI RMF).
11. European Union, Artificial Intelligence Act.
12. Huw Roberts, Mariarosaria Taddeo, and Luciano Floridi, "A Framework for Evaluating Global AI Governance Initiatives," Global Policy, 2026.
13. Inioluwa Deborah Raji et al., "The Fallacy of AI Functionality," Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 2022.
14. Donghee Shin, "Designing Nudges, Designing Trust: Toward Ethical and Human-Centered AI," in Human-Centered AI: Advancing Research and Applications in Human-AI Interaction, Springer Nature Singapore, 2026.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Deep Learning

Course Type: Lab/Project

Course Objectives

- To understand the fundamental concepts and algorithms of Reinforcement Learning and Deep Reinforcement Learning.
- To apply Deep Reinforcement Learning techniques for solving complex sequential decision-making problems.
- To develop and evaluate DRL-based solutions for real-world intelligent applications.

Course Outcomes

CO1	Understand the fundamental principles of Reinforcement Learning.
CO2	Apply classical Reinforcement Learning algorithms for sequential decision-making problems.
CO3	Implement and evaluate Deep Reinforcement Learning algorithms using deep neural networks.
CO4	Design and analyze Deep Reinforcement Learning-based solutions for real-world applications.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	1	2	2	1
CO2	3	3	2	2	3	2
CO3	3	3	3	2	3	3
CO4	3	3	3	3	3	3

Syllabus

Introduction to Reinforcement Learning: intelligent agents, environment, states, actions, rewards, episodes, return, and discount factor. Exploration and exploitation trade-off. Markov Decision Process (MDP): policy, value function, action-value function, and Bellman equations. Dynamic Programming approaches: policy iteration and value iteration. Model-free Reinforcement Learning: Monte Carlo methods, Temporal Difference (TD) learning, SARSA, Q-Learning, and function approximation techniques.

Introduction to Deep Reinforcement Learning (DRL): integration of deep learning and reinforcement learning. Deep neural networks for reinforcement learning. Deep Q-Networks (DQN): architecture, experience replay, target networks, and training process. Advanced value-based methods: Double DQN and Dueling DQN. Policy-based learning approaches. Actor-Critic architectures. Neural network architectures for DRL: Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Training challenges, hyperparameter tuning, reward design, and performance optimization.

Advanced DRL algorithms: Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Critic (SAC). On-policy and off-policy learning strategies.

Introduction to Multi-Agent Reinforcement Learning (MARL): concepts, coordination, and challenges. Applications of Deep Reinforcement Learning in robotics, autonomous systems, games, healthcare, finance, recommendation systems, and other real-world domains. Case studies and analysis of DRL-based solutions.

Text Books/References

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, MIT Press, Cambridge, MA, 2018.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, Cambridge, MA, 2016.
3. Laura Graesser and Wah Loon Keng, Foundations of Deep Reinforcement Learning: Theory and Practice in Python, Addison-Wesley Professional, 2022.
4. Maxim Lapan, Deep Reinforcement Learning Hands-On: Apply Modern Deep Reinforcement Learning Algorithms with PyTorch and Gym, 2nd Edition, Packt Publishing, Birmingham, UK, 2020.
5. Sudharsan Ravichandiran, Hands-On Reinforcement Learning with Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow, Packt Publishing, Birmingham, UK, 2020.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Computer Networks and Wireless Communications

Course Type: Lab

Course Objectives

- To introduce the evolution, architecture, and key technologies of modern mobile communication networks from 4G LTE to 5G and beyond.
- To provide an understanding of advanced wireless technologies including small cells, cognitive networks, SDN, Open RAN, and intelligent network architectures.
- To enable students to analyze and design mobile network solutions for emerging applications such as IoT, edge computing, and industry-specific 5G use cases.

Course Outcomes

CO1	Understand the evolution, architecture, protocols, and mobility management mechanisms of 4G LTE/LTE-Advanced networks.
CO2	Analyze small cell networks, network convergence approaches, cognitive networks, and AI/ML-driven wireless communication systems.
CO3	Evaluate 5G system architecture, key technology enablers, Open RAN, SDN-based network management, and edge computing frameworks.
CO4	Design and assess 5G-based solutions for emerging applications including IoT, cyber-physical systems, smart cities, autonomous systems, and industry verticals.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	1		2	
CO2	2	3	2	1	2	1
CO3	2	3	2	2	3	1
CO4	2	2	3	2	3	2

Syllabus

Evolution of mobile communication systems and 4G LTE/LTE-Advanced technologies. Small cell networks: concepts, deployment strategies, and use cases. 4G network architecture: Access Stratum (AS) and Non-Access Stratum (NAS). LTE Radio Access Network (RAN): physical layer, protocols, and architecture. Key LTE network and User Equipment (UE) procedures: call setup and release, mobility management in idle mode, active mode mobility, and handover mechanisms.

Convergence of wireless networks: unlicensed spectrum, private networks, neutral host networks, and Wi-Fi technology evolution. LTE-Wi-Fi integration and heterogeneous network architectures. Software Defined Radio (SDR): concepts, architecture, and applications. Cognitive radio and AI/ML-driven cognitive networks: intelligent spectrum management, adaptive communication, and self-optimizing networks.

Introduction to 5G systems: network of networks, devices, systems, machines, and Big Data-driven communication. Key technology enablers for 5G architecture: enhanced mobile broadband (eMBB),

massive IoT (mMTC), and ultra-reliable low-latency communication (URLLC). 5G edge computing architecture supporting massive IoT devices and Cyber Physical Systems. Software Defined Networking (SDN) and network performance management. Advances in Radio Access Networks (RAN): 5G Open RAN architecture, RAN Intelligent Controller (RIC), xApps, rApps, and dApps. Frameworks for drive testing, log analysis, network diagnostics, and performance evaluation. Case studies of 5G applications in healthcare, enterprise networks, autonomous vehicles, and smart cities, focusing on protocol stacks, latency requirements, and high data-rate communication.

Text Books/References

1. Albert-László Barabási, Network Science, 1st Edition, Cambridge University Press, 2016.
2. Matthew A. Russell, Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More, 3rd Edition, O'Reilly Media, 2018.
3. William L. Hamilton, Graph Representation Learning, 1st Edition, Morgan & Claypool Publishers, 2020.
4. Dean Allemang and James Hendler, Semantic Web for the Working Ontologist: Effective Modelling in RDFS and OWL, 2nd Edition, Morgan Kaufmann, 2011.
5. Lei Tang and Huan Liu, Community Detection and Mining in Social Media, 1st Edition, Morgan & Claypool Publishers, 2010.
6. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, 1st Edition, Cambridge University Press, 2010.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Linear Algebra, Basic Probability and Statistics, Python Programming (preferred)

Course Type: Lab/Project

Course Objectives

- To introduce the fundamental concepts of image processing and computer vision techniques for visual data analysis.
- To provide an understanding of classical and deep learning-based approaches for image and video understanding.
- To enable students to develop and evaluate computer vision solutions for real-world applications.

Course Outcomes

CO1	Understand the fundamentals of image formation, representation, enhancement, and processing techniques used in computer vision.
CO2	Apply image analysis, segmentation, and feature extraction techniques for solving computer vision problems.
CO3	Analyze and utilize deep learning approaches including CNNs, transfer learning, attention mechanisms, and Vision Transformers for visual recognition tasks.
CO4	Design and evaluate computer vision applications using object detection, tracking, video analysis, multimodal vision, and edge-based vision systems.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	1	2	1	
CO2	3	3	2	2	3	1
CO3	3	3	3	2	3	2
CO4	3	3	3	3	3	2

Syllabus

Introduction to Computer Vision and its applications. Image formation, image sensing, sampling, quantization, and digital image representation. Image enhancement techniques: histogram processing, interpolation, spatial filtering, smoothing, sharpening, and noise reduction. Image quality assessment and evaluation metrics. Image transformations: Fourier Transform, Discrete Cosine Transform (DCT), and overview of wavelet transforms. Image restoration, frequency-domain filtering, and mathematical morphology.

Image segmentation techniques: thresholding, edge detection, region-based segmentation, watershed algorithm, and evaluation metrics. Image representation and feature extraction techniques. Color models, texture analysis, and feature descriptors including SIFT, SURF, ORB, and HOG. Feature matching and object recognition using classical computer vision approaches. Applications of image analysis in healthcare, autonomous systems, agriculture, and surveillance.

Introduction to deep learning-based computer vision. Convolutional Neural Networks (CNNs),

convolution operations, pooling, feature learning, and image classification. Transfer learning and pretrained models. Attention mechanisms and Vision Transformers (ViT). Self-supervised learning and vision foundation models. Video representation and analysis: optical flow, object detection, object tracking, pose estimation, and human activity recognition. Multimodal vision systems: image captioning, Vision-Language Models, Segment Anything Models (SAM), and Edge AI-based vision applications. Case studies on intelligent vision systems.

Text Books/References

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018.
2. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, Springer, 2022.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
4. Mohamed Elgendy, Deep Learning for Vision Systems, Manning Publications, 2020.
5. Joseph Howse and Joe Minichino, Learning OpenCV 4 Computer Vision with Python, Packt Publishing, 2020.
6. Hands-On Vision Transformers with PyTorch, Packt Publishing, 2024.
7. Recent research articles from reputed journals and conferences in the field of Computer Vision and Artificial Intelligence.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Fundamentals of Artificial Intelligence and Machine Learning

Course Type: Lab

Course Objectives

- To introduce knowledge representation, semantic technologies, and ontology-based knowledge systems for intelligent applications.
- To provide an understanding of knowledge graph engineering, semantic retrieval, and knowledge-driven AI systems.
- To explore the integration of knowledge systems with Large Language Models and emerging Semantic AI approaches.

Course Outcomes

CO1	Understand knowledge representation techniques, semantic web technologies, ontologies, and reasoning mechanisms for knowledge systems.
CO2	Design and develop knowledge graphs using ontology engineering, graph databases, information extraction, and semantic retrieval techniques.
CO3	Apply knowledge-based approaches for semantic search, interoperability, and intelligent applications across different domains.
CO4	Analyze and develop Semantic AI solutions using knowledge-enhanced LLMs, Retrieval-Augmented Generation, and neuro-symbolic AI approaches.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	1		2	1
CO2	2	3	3	2	3	1
CO3	2	3	2	2	3	2
CO4	2	3	3	2	3	2

Syllabus

Evolution from Semantic Web to Semantic AI. Knowledge representation paradigms. Resource Description Framework (RDF) and RDF*, RDF Schema (RDFS), Web Ontology Language (OWL), and Description Logics. Ontology engineering methodologies, ontology design patterns, ontology development tools, and semantic reasoning. SPARQL query language and semantic interoperability through Linked Open Data.

From ontologies to knowledge graphs. Knowledge graph fundamentals and architecture. Triple stores and graph databases. Knowledge graph construction: information extraction, entity recognition, entity linking, relation extraction, and knowledge graph population. Semantic search, knowledge graph embeddings, vector databases, enterprise knowledge graphs, and applications in healthcare, agriculture, Industry 4.0, smart cities, and scientific knowledge management.

Introduction to Large Language Models (LLMs) and limitations of standalone LLMs. Knowledge-augmented LLMs, Retrieval-Augmented Generation (RAG), Graph RAG, and LLM–Knowledge Graph integration. Neuro-symbolic AI, semantic agents, explainable AI using knowledge graphs, trustworthy AI, and future directions in Semantic AI.

Text Books/References

1. Dean Allemang, Jim Hendler, and Fabien Gandon, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, 3rd Edition, Association for Computing Machinery, 2020.
2. Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph, Foundations of Semantic Web Technologies, Chapman & Hall/CRC, 2009.
3. Tom Heath and Christian Bizer, Linked Data: Evolving the Web into a Global Data Space, Morgan & Claypool Publishers, 2011.
4. Grigoris Antoniou and Frank van Harmelen, A Semantic Web Primer, 4th Edition, MIT Press, 2023.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Pre-Requisite(s): Fundamentals of Artificial Intelligence and Machine Learning

Course Type: Lab

Course Objectives

- Understand the principles and methodologies of evolutionary intelligence for optimization and machine learning.
- Analyze advanced evolutionary learning paradigms including neuro-evolution, adversarial learning, and co-evolution for adaptive intelligent systems.
- Apply evolutionary techniques to automated algorithm design, AutoML, and many/multi-objective optimization problems.
- Evaluate open-ended evolutionary systems, quality-diversity methods, and explainability frameworks for trustworthy evolving intelligence.

Course Outcomes

CO1	Understand the principles of evolving intelligence, including evolutionary computation, neuro-evolution, and adaptive learning paradigms.
CO2	Apply evolutionary techniques for AutoML, automated algorithm design, and multi-/many-objective optimization problems.
CO3	Analyze quality-diversity methods, open-ended evolution, and optimization approaches using suitable evaluation metrics.
CO4	Design and implement adaptive, interpretable, and trustworthy intelligent systems by integrating evolutionary intelligence with XAI and XEC.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	1		2	1
CO2	2	3	3	2	3	1
CO3	2	3	2	2	3	2
CO4	2	3	3	2	3	2

Syllabus

Introduction to Evolving Intelligence; Evolutionary Computation as a Learning Paradigm; Fundamentals of Evolutionary Computation; Search, Optimization, and Adaptation; Evolution as a Search Process; Evolutionary Search in Intelligent Systems; Evolution Strategies in Modern Artificial Intelligence; Quality Diversity Optimization; Novelty Search; Behaviour Characterization; MAP-Elites; Illumination Algorithms; Diversity Preservation Techniques; Fundamentals of Multi-objective Optimization; Pareto Dominance and Pareto Fronts; Many-objective Optimization; Evolutionary Multi-objective Algorithms; Decomposition-based and Indicator-based Methods; Constraint Handling Techniques; Applications in Engineering Design and Machine Learning.

Automated Algorithm Design; Algorithm Configuration and Selection; Hyperparameter Optimization; Hyper-heuristics; Evolutionary Program Synthesis; Automated Machine Learning (AutoML); Evolutionary AutoML Pipelines; Fundamentals of Neuro-evolution; Evolution of Neural Network Weights and Architectures; NeuroEvolution of Augmenting Topologies (NEAT); Evolutionary Neural Architecture Search (NAS); Synergies between Evolutionary Intelligence and

Generative AI; Fundamentals of Adversarial Learning; Evolutionary Adversarial Optimization; Evolutionary Generative Adversarial Networks; Robust Learning through Evolution; Competitive and Cooperative Co-evolution; Multi-agent Evolutionary Learning; Applications in Robotics, Game AI, and Adaptive Intelligent Systems.

Open-ended Evolution; Artificial Life and Open-ended Learning; Lifelong and Continual Learning; Autonomous Skill Discovery; Explainable Evolving Intelligence; Explainable Artificial Intelligence (XAI); Explainable Evolutionary Computation (XEC); Interpretation of Evolutionary Models; Explainability in Neuro-evolution; Trustworthy, Ethical, and Responsible Evolving Intelligence; Emerging Research Trends; Industrial and Scientific Applications of Evolving Intelligence.

Text Books/References

1. A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, 2nd Edition, Springer, 2015.
2. Kenneth A. De Jong, Evolutionary Computation: A Unified Approach, 2nd Edition, MIT Press, 2024.
3. Sebastian Risi, Yujin Tang, David Ha, and Risto Miikkulainen, Neuroevolution: Harnessing Creativity in AI Agent Design, MIT Press, 2025.
4. Bastien Chopard and Marco Tomassini, An Introduction to Metaheuristics for Optimization, 2nd Edition, Springer, 2026.
5. Wolfgang Banzhaf, Penousal Machado, and Mengjie Zhang (Eds.), Handbook of Evolutionary Machine Learning, Springer, 2024.
6. Hasmat Malik, Yashwant Singh, Rajesh Kumar Dhanaraj, Ponnambalam Kumar, and Tuan Anh Nguyen (Eds.), Metaheuristic and Evolutionary Computation: Algorithms and Applications, Springer, 2021.
7. Xin-She Yang, Nature-Inspired Optimization Algorithms, 2nd Edition, Elsevier, 2021.
8. Mansour Eddaly, Bassem Jarboui, and Patrick Siarry (Eds.), Metaheuristics for Machine Learning: New Advances and Tools, Springer, 2023.
9. Dhish Kumar Saxena, Kalyanmoy Deb, and others, Machine Learning Assisted Evolutionary Multi- and Many-Objective Optimization, Springer, 2024.
10. Ke-Lin Du and M. N. S. Swamy, Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature, Springer, 2016.
11. Recent research articles from leading journals and conferences in Evolutionary Computation, Artificial Intelligence, Machine Learning, and Optimization.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	20	
Continuous Assessment (Lab)	30	
End Semester		30

Note: Continuous assessment may include quizzes, tutorials, laboratory assessments, case studies, assignments, and other course-related evaluation components. The Mid-Term examination may be conducted as a two-hour written examination or a two-hour laboratory-integrated examination, as applicable. The End-Semester assessment may be conducted either as a two-hour written examination, a laboratory-integrated examination, or as a project presentation and evaluation, provided the project-based assessment is approved by the Department.

Prerequisite: An open mind and the urge for self-development, basic English language skills and knowledge of high school level arithmetic.

Course Objectives:

- Help students transit from campus to corporate and enhance their soft skills
- Enable students to understand the importance of goal setting and time management skills
- Support them in developing their problem solving and reasoning skills
- Inspire students to enhance their diction, grammar and verbal reasoning skills

Course Outcomes:

CO1	Soft Skills - To develop positive mindset, communicate professionally, manage time effectively and set personal goals and achieve them.
CO2	Soft Skills - To make formal and informal presentations with self-confidence.
CO3	Aptitude - To analyze, understand and employ the most suitable methods to solve questions on arithmetic and algebra.
CO4	Aptitude - To analyze, understand and apply suitable techniques to solve questions on logical reasoning and data analysis.
CO5	Verbal - To infer the meaning of words and use them in the right context. To have a better understanding of the nuances of English grammar and become capable of applying them effectively.
CO6	Verbal - To identify the relationship between words using reasoning skills. To understand and analyze arguments and use inductive/deductive reasoning to arrive at conclusions and communicate ideas/perspectives convincingly.

CO-PO Mapping

CO	PO1	PO2	PO3
CO1	2	1	-
CO2	2	1	-
CO3	2	1	-
CO4	2	1	-
CO5	1	2	-
CO6	2	2	-

Syllabus

Soft Skills

Introduction to 'campus to corporate transition':

Communication and listening skills: communication process, barriers to communication, verbal and non-verbal communications, elements of effective communication, listening skills, empathetic listening, role of perception in communication.

Assertiveness skills: the concept, assertiveness and self-esteem, advantages of being assertive, assertiveness and organizational effectiveness.

Self-perception and self-confidence: locus of control (internal v/s external), person perception, social perception, attribution theories-self presentation and impression management, the concept of self and self-confidence, how to develop self-confidence.

Goal setting: the concept, personal values and personal goals, goal setting theory, six areas of goal setting, process of goal setting: SMART goals, how to set personal goals

Time management: the value of time, setting goals/ planning and prioritizing, check the time killing habits, procrastination, tools for time management, rules for time management, strategies for effective time management

Presentation skills: the process of presentation, adult learning principles, preparation and planning, practice, delivery, effective use of voice and body language, effective use of audio-visual aids, dos and don'ts of effective presentation

Public speaking-an art, language fluency, the domain expertise (Business GK, Current affairs), self-confidence, the audience, learning principles, body language, energy level and conviction, student presentations in teams of five with debriefing

Verbal

Vocabulary: Familiarize students with the etymology of words, help them realize the relevance of word analysis and enable them to answer synonym and antonym questions. Create an awareness about the frequently misspelt words, commonly confused words and wrong form of words in English.

Grammar: Train students to understand the nuances of English Grammar and thereby enable them to spot grammatical errors and punctuation errors in sentences.

Reasoning: Stress the importance of understanding the relationship between words through analogy questions and learn logical reasoning through syllogism questions. **Emphasize the importance of avoiding the gap (assumption) in arguments/ statements/ communication.**

Oral Communication Skills: Aid students in using the gift of the gap to improve their debating skills.

Writing Skills: Introduce formal written communication and keep the students informed about the etiquettes of email writing. Make students practice writing emails, especially composing job application emails.

Aptitude

Numbers: Types, Power Cycles, Divisibility, Prime, Factors & Multiples, HCF & LCM, Surds, Indices, Square roots, Cube Roots and Simplification.

Percentage: Basics, Profit, Loss & Discount, and Simple & Compound Interest.

Ratio, Proportion & Variation: Basics, Alligations, Mixtures, and Partnership.

Averages: Basics, and Weighted Average.

Time and Work: Basics, Pipes & Cistern, and Work Equivalence.

Time, Speed and Distance: Basics, Average Speed, Relative Speed, Boats & Streams, Races and Circular tracks.

Statistics: Mean, Median, Mode, Range, Variance, Quartile Deviation and Standard Deviation.

Data Interpretation: Tables, Bar Diagrams, Line Graphs, Pie Charts, Caselets, Mixed Varieties, and other forms of data representation.

Equations: Basics, Linear, Quadratic, Equations of Higher Degree and Problems on ages.

Logarithms, Inequalities and Modulus: Basics

References

Soft Skills:

Communication and Listening Skills

1. DuBrin, A. J. Applied Psychology: Individual and Organizational Effectiveness. Pearson Prentice Hall, 2004.
2. Aamodt, M. G. Industrial/Organizational Psychology: An Applied Approach. 6th ed., Wadsworth Cengage Learning, 2010.

Assertiveness Skills

3. Bolton, R., & Bolton, D. G. People Styles at Work... and Beyond: Making Bad Relationships Good and Good Relationships Better. Ridge Associates Inc., 2009.
4. Hayes, J. Interpersonal Skills at Work. 2nd ed., Routledge, 2003.
5. Nord, W. R., Brief, A. P., Atieh, J. M., & Doherty, E. M. Meanings of Occupational Work: A Collection of Essays (pp. 21–64). Lexington Books, 1990.

Self-Perception and Self-Confidence

6. Martinko, M. J. Attribution Theory: An Organizational Perspective. St. Lucie Press, 1995.
7. Hewstone, M. Attribution Theory: Social and Functional Extensions. Blackwell Publishers, 1983.

Time Management

8. Covey, S. R. The 7 Habits of Highly Effective People. Revised ed., Free Press, 2004.
9. Blanchard, K. H. The 25 Best Time Management Tools & Techniques: How to Get More Done Without Driving Yourself Crazy. 1st ed., Peak Performance Press, 2005.
10. Blanchard, K. H., & Johnson, S. The One Minute Manager. William Morrow, 1984.

Verbal:

1. Meltzer, E. The Ultimate Guide to SAT Grammar.
2. Green, S., & Wolf, I. K. Barron's New GRE. Barron's Educational Series, 2011.
3. Kolby, J., Thornburg, S., & Pierce, K. Nova's GRE Prep Course.
4. Kaplan. Kaplan New GRE Premier. 2011–2012.
5. Kaplan. Kaplan's GRE Comprehensive Program.
6. Lewis, N. Word Power Made Easy. Goyal Publishers, Reprint ed., 2011.
7. Manhattan Prep. GRE Verbal Strategies: Effective Strategies & Practice from 99th Percentile Instructors.
8. Pearson. A Complete Manual for CAT. 2013.
9. Aggarwal, R. S. A Modern Approach to Verbal Reasoning.
10. Upendran, S. Know Your English. Universities Press (India) Limited, 2015.
11. Green, S. W., & Wolf, I. K. Barron's GRE. 19th ed., Barron's Educational Series, 2019.
12. Wren, P. C., & Martin, H. High School English Grammar & Composition.
13. BBC Learning English. <https://www.bbc.co.uk/learningenglish>
14. Cambridge English. <https://www.cambridgeenglish.org>
15. English for Everyone. <https://www.englishforeveryone.org>
16. Merriam-Webster Dictionary. <https://www.merriam-webster.com>

Aptitude:

1. Sharma, A. How to Prepare for Quantitative Aptitude for the CAT: Common Admission Test. 5th ed., Tata McGraw-Hill, 2012.
2. Sharma, A. How to Prepare for Logical Reasoning for the CAT: Common Admission Test. 2nd ed., Tata McGraw-Hill, 2014.
3. Sharma, A. How to Prepare for Data Interpretation for the CAT: Common Admission Test. 3rd ed., Tata McGraw-Hill, 2015.
4. Aggarwal, R. S. Quantitative Aptitude for Competitive Examinations. S. Chand Publishing, 2015.
5. Aggarwal, R. S. A Modern Approach to Verbal & Non-Verbal Reasoning. Revised ed., S. Chand Publishing, 2015.
6. Verma, S. Quantum CAT: Quantitative Aptitude. Arihant Publications, 2016.
7. MBAtious. <https://www.mbatious.com>
8. CampusGate. <https://www.campusgate.co.in>

9. Career Bless. <https://www.careerbless.com>

Evaluation Pattern

Assessment	Internal	External
Continuous Assessment (CA)* – Soft Skills	30	-
Continuous Assessment (CA)* – Aptitude	10	25
Continuous Assessment (CA)* – Verbal	10	25
Total	50	50
Pass / Fail		

*CA - Can be presentations, speaking activities and tests.

Pre-requisite: Willingness to learn, team spirit, basic English language and communication skills and knowledge of high school level arithmetic.

Course Objectives:

- Help students to understand the importance of interpersonal skills and team work
- Prepare the students for effective group discussions and interviews participation.
- Help students to sharpen their problem solving and reasoning skills
- Empower students to communicate effectively by using the correct diction, grammar and verbal reasoning skills

Course Outcomes:

CO1	Soft Skills - To demonstrate good interpersonal skills, solve problems and effectively participate in group discussions.
CO2	Soft Skills - To write technical resume and perform effectively in interviews.
CO3	Aptitude - To identify, investigate and arrive at appropriate strategies to solve questions on arithmetic by managing time effectively.
CO4	Aptitude - To investigate, understand and use appropriate techniques to solve questions on logical reasoning and data analysis by managing time effectively.
CO5	Verbal - To be able to use diction that is more refined and appropriate and to be competent in knowledge of grammar to correct/improve sentences.
CO6	Verbal - To be able to examine, interpret and investigate passages and to be able to generate ideas, structure them logically and express them in a style that is comprehensible to the audience/recipient.

CO-PO Mapping

PO/CO	PO1	PO2	PO3
CO1	2	1	-
CO2	2	1	-
CO3	2	1	-
CO4	2	1	-
CO5	1	2	-
CO6	2	2	-

Syllabus

Soft Skills

Interpersonal skill: ability to manage conflict, flexibility, empathetic listening, assertiveness, stress management, problem solving, understanding one's own interpersonal needs, role of effective team work in organizations

Group problem solving: the process, the challenges, the skills and knowledge required for the same.

Conflict management: the concept, its impact and importance in personal and professional lives, (activity to identify personal style of conflict management, developing insights that helps in future conflict management situations.)

Team building and working effectively in teams: the concept of groups (teams), different stages

of group formation, process of team building, group dynamics, characteristics of effective team, role of leadership in team effectiveness. (Exercise to demonstrate the process of emergence of leadership in a group, debrief and reflection), group discussions.

Interview skills: what is the purpose of a job interview, types of job interviews, how to prepare for an interview, dos and don'ts of interview, One on one mock interview sessions with each student

Verbal

Vocabulary: Help students understand the usage of words in different contexts. Stress the importance of using refined language through idioms and phrasal verbs.

Grammar: Enable students to identify poorly constructed sentences or incorrect sentences and improvise or correct them.

Reasoning: Facilitate the student to tap her/his reasoning skills through critical reasoning questions and logical ordering of sentences.

Reading Comprehension: Enlighten students on the different strategies involved in tackling reading comprehension questions.

Public Speaking Skills: Empower students to overcome glossophobia and speak effectively and confidently before an audience.

Writing Skills: Practice closet tests that assess basic knowledge and skills in usage and mechanics of writing such as punctuation, basic grammar and usage, sentence structure and rhetorical skills such as writing strategy, organization, and style.

Aptitude

Sequence and Series: Basics, AP, GP, HP, and Special Series.

Geometry: 2D, 3D, Coordinate Geometry, and Heights & Distance.

Permutations & Combinations: Basics, Fundamental Counting Principle, Circular Arrangements, and Derangements.

Probability: Basics, Addition & Multiplication Theorems, Conditional Probability and Bayes' Theorem.

Logical Reasoning I: Arrangements, Sequencing, Scheduling, Venn Diagram, Network Diagrams, Binary Logic, and Logical Connectives, Clocks, Calendars, Cubes, Non-Verbal reasoning and Symbol based reasoning.

Logical Reasoning II: Blood Relations, Direction Test, Syllogisms, Series, Odd man out, Coding & Decoding, Cryptarithmic Problems and Input - Output Reasoning.

Data Sufficiency: Introduction, 5 Options Data Sufficiency and 4 Options Data Sufficiency.

Campus recruitment papers: Discussion of previous year question papers of all major recruiters of Amrita Vishwa Vidyapeetham.

Miscellaneous: Interview Puzzles, Calculation Techniques and Time Management Strategies.

References

Soft Skills:

Team Building

1. Quick, T. L. Successful Team Building. AMACOM, 1992.
2. Miller, B. C. Quick Team-Building Activities for Busy Managers: 50 Exercises That Get Results in Just 15 Minutes. 1st ed., AMACOM, 2003.
3. Lencioni, P. The Five Dysfunctions of a Team: A Leadership Fable. 1st ed., Jossey-Bass, 2002.

Verbal:

1. Graduate Management Admission Council (GMAC). GMAT Official Guide. 2019.

2. Sharma, A. How to Prepare for Verbal Ability and Reading Comprehension for CAT.
3. Meissner, J. Turbocharge Your GMAT Sentence Correction Study Guide. 2012.
4. Kaplan. Kaplan GMAT 2012–13. Kaplan Publishing, 2012.
5. Kaplan. New GMAT Premier. Kaplan Publishing, 2013.
6. Manhattan Prep. Critical Reasoning GMAT. 6th ed.
7. Manhattan Prep. Sentence Correction GMAT. 6th ed.
8. Barrett, M. SAT Prep Black Book: The Most Effective SAT Strategies Ever Published.
9. Bryon, M. Verbal Reasoning Test Workbook: Unbeatable Practice for Verbal Ability, English Usage, Interpretation and Judgement Tests.
10. University of Bristol – Grammar Tutorial. https://www.bristol.ac.uk/arts/skills/grammar/grammar_tutorial/page_55.htm
11. CampusGate. <https://www.campusgate.co.in>

Aptitude:

1. Sharma, A. How to Prepare for Quantitative Aptitude for the CAT: Common Admission Test. 5th ed., Tata McGraw-Hill, 2012.
2. Sharma, A. How to Prepare for Logical Reasoning for the CAT: Common Admission Test. 2nd ed., Tata McGraw-Hill, 2014.
3. Sharma, A. How to Prepare for Data Interpretation for the CAT: Common Admission Test. 3rd ed., Tata McGraw-Hill, 2015.
4. Aggarwal, R. S. Quantitative Aptitude for Competitive Examinations. S. Chand Publishing, 2015.
5. Aggarwal, R. S. A Modern Approach to Verbal & Non-Verbal Reasoning. Revised ed., S. Chand Publishing, 2015.
6. Verma, S. Quantum CAT: Quantitative Aptitude. Arihant Publications, 2016.
7. MBAtious. <https://www.mbatious.com>
8. CampusGate. <https://www.campusgate.co.in>
9. Career Bless. <https://www.careerbless.com>

Evaluation Pattern

Assessment	Internal	External
Continuous Assessment (CA)* – Soft Skills	30	-
Continuous Assessment (CA)* – Aptitude	10	25
Continuous Assessment (CA)* – Verbal	10	25
Total	50	50

*CA - Can be presentations, speaking activities and tests.

25AVP501

Mastery Over Mind (MAOM)

1-0-2 2

Course Outcomes

CO01: Relate to the causes of stress in one's life.

CO02: Experiment with a range of relaxation techniques

CO03: Model a meditative approach to work, study, and life.

CO04: Develop appropriate practice of MA-OM technique that is effective in one's life

CO05: Inculcate a higher level of awareness and focus.

CO06: Evaluate the impact of a meditation technique

1. Course Overview

Master Over the Mind (MAOM) is an Amrita initiative to implement schemes and organise university-wide programs to enhance health and wellbeing of all faculty, staff, and students (UN SDG -3). This program as part of our efforts for sustainable stress reduction gives an introduction to immediate and long-term benefits and equips every attendee to manage stressful emotions and anxiety facilitating inner peace and harmony.

With a meditation technique offered by Amrita Chancellor and world-renowned humanitarian and spiritual leader, Sri Mata Amritanandamayi Devi (Amma), this course has been planned to be offered to all students of all campuses of AMRITA, starting off with all first years, wherein one hour per week is completely dedicated for guided practical meditation session and one hour on the theory aspects of MAOM. The theory section comprises lecture hours within a structured syllabus and will include invited guest lecture series from eminent personalities from diverse fields of excellence. This course will enhance the understanding of experiential learning based on university's mission: "Education for Life along with Education for Living", and is aimed to allow learners to realize and rediscover the infinite potential of one's true Being and the fulfilment of life's goals.

2. Course Syllabus

Unit 1 (4 hours)

Causes of Stress: The problem of not being relaxed. Need for meditation -basics of stress management at home and workplace. Traditions and Culture. Principles of meditation– promote a sense of control and autonomy in the Universal Human Value System. Different stages of Meditation. Various Meditation Models. Various practices of Meditation techniques in different schools of philosophy and Indian Knowledge System.

Unit 2 (4 hours)

Improving work and study performance. Meditation in daily life. Cultivating compassion and good mental health with an attitude of openness and acceptance. Research and Science of Meditation: Significance of practising meditation and perspectives from diverse fields like science, medicine, technology, philosophy, culture, arts, management, sports, economics, healthcare, environment etc. The role of meditation for stress and anxiety reduction in one's life with insights based on recent cutting-edge technology. The effect of practicing meditation for the wholesome wellbeing of an individual.

Unit 3 (4 hours)

Communications: principles of conscious communication. Relationships and empathy: meditative approach in managing and maintaining better relationships in life during the interactions in the world, role of MAOM in developing compassion, empathy and responsibility, instilling interest, and orientation to humanitarian projects as a key to harness intelligence and compassion in youth. Methodologies to evaluate effective awareness and relaxation gained from meditation. Evaluating the global transformation through meditation by instilling human values which leads to service learning and compassion driven research.

TEXT BOOKS:

- 1.Mata Amritanandamayi Devi, "Cultivating Strength and vitality," published by Mata Amritanandamayi Math, Dec 2019
- 2.Swami Amritaswarupananda Puri, "The Color of Rainbow " published by MAM, Amritapuri.

REFERENCES:

- 1.Craig Groeschel, "Winning the War in Your Mind: Change Your Thinking, Change Your Life" Zondervan Publishers, February 2019
- 2.R Nagarathna et al, "New Perspectives in Stress Management "Swami Vivekananda Yoga Prakashana publications, Jan 1986
3. Swami Amritaswarupananda Puri "Awaken Children Vol 1, 5 and 7 - Dialogues with Amma on Meditation", August 2019

4. Swami Amritaswarupananda Puri “From Amma’s Heart - Amma’s answer to questions raised during world tours” March 2018
5. Secret of Inner Peace- Swami Ramakrishnananda Puri, Amrita Books, Jan 2018.
6. Mata Amritanandamayi Devi “Compassion :The only way to Peace:Paris Speech”, MA Center, April 2016.
7. Mata Amritanandamayi Devi “Understanding and collaboration between Religions”, MA Center, April 2016.
8. Mata Amritanandamayi Devi “Awakening of Universal Motherhood: Geneva Speech” M A center, April 2016.

Course Overview

The course introduces students to the Integrated Amrita Meditation Technique (IAM 20), a structured practice that combines yogic stretches, regulated breathing (pranayama), and guided meditation. The course integrates both theoretical understanding and experiential practice to enhance emotional balance, stress resilience, concentration, and overall holistic development.

Aligned with United Nations Sustainable Development Goal 3 (SDG 3): Good Health and Well-Being, the course promotes preventive mental health care, emotional well-being, and sustainable lifestyle practices among students. By fostering inner stability, self-awareness, and self-regulation, the programme supports academic excellence and contributes to the development of healthy individuals and communities.

The course is expected to improve student well-being, enhance academic focus and cognitive clarity, reduce stress-related indicators, and promote compassionate and emotionally supportive culture.

Course Outcomes

CO1	Explain principles and components of IAM 20.
CO2	Demonstrate and correct meditation practice independently.
CO3	Analyze personal stress patterns using meditation insights.
CO4	Evaluate personal growth through daily practice.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6
CO						
CO1			1			
CO2			1			
CO3			1	1		
CO4			1			

1- Weak affinity, 2-Moderate affinity, 3-Strong affinity

Syllabus

Unit 1: Foundations of Meditation (CO1)

- Meaning and purpose of meditation
- Demonstration and supervised training
- Overview of structure of IAM
- Guidance for safe and effective practice
- Importance of mental health

Unit 2: Theoretical Foundations (CO2)

- Structure and components of IAM: yogic stretches, regulated breathing, and guided meditation
- Breath–mind connection
- Integration of IAM practice into daily routine and academic life

- Application of IAM for stress management, concentration, and holistic student development

Unit 3: IAM Practical Training (CO3)

- Preparatory stretches
- Pranayama components
- Guided meditation sequence
- Corrections and common challenges

Unit 4: Integration and Transformation (CO4)

- Building daily discipline
- Research evidence in higher education
- Compassion and mindful communication

Evaluation Pattern

Assessment	Internal (60)	External (40)
Attendance	20 Marks	
Class Participation	20 Marks	
Activity	20 Marks	
Practical Demonstration		30 Marks
Viva		10 Marks

Lecture Plan

Lectures	Topics
Week 1	Introduction to IAM, signing agreement form, conduct Pre Test
Week 2	Practice of Yogic Stretches
Week 3	Practice of Yogic Stretches
Week 4	Practice of Deep Breathing, Visualization, and Meditation
Week 5	Practice of Full Cycle of IAM 20
Week 6	Mid-Intervention data collection
Week 7	Practice of Full Cycle of IAM 20
Week 8 and Week 9	Evaluation Activities: (Faculty can Decide) Poster Presentation, Viva, Role Play, Paper Presentation, or Video Presentation
Week 10	Conduct Post Test and Feedback
Week 11	Practice of Full Cycle of IAM 20
Week 12	Practice of Full Cycle of IAM 20

CIR Course – PG – Paper I

Pre-requisite:

An open mind and the urge for self-development, basic English language skills and knowledge of high school level arithmetic.

Course Objectives:

- To strengthen professional skills for academic and research environments.
- To enhance presentation skills for seminars, reviews, and academic discussions.
- To guide students in effective goal setting and time management for postgraduate studies.
- Support them in developing their problem solving and reasoning skills
- Inspire students to enhance their diction, grammar, verbal reasoning and communication skills

Course Outcomes:

CO1: Soft Skills - To develop a positive mindset, communicate professionally, manage time effectively, set personal goals and achieve them.

CO2: Soft Skills - To make formal and informal presentations with self-confidence.

CO3: Aptitude - To analyze, understand and employ the most suitable methods to solve questions on arithmetic and algebra.

CO4: Aptitude - To analyze, understand and apply suitable techniques to solve questions in data analysis.

CO5: Verbal - To infer the meaning of words and use them in the right context. To have a clearer understanding of the nuances of English grammar and become capable of applying them effectively.

CO6: Verbal - To identify the relationship between words using reasoning skills. To understand and analyze arguments and use verbal reasoning skills to arrive at conclusions and communicate ideas/perspectives convincingly.

CO-PO Mapping - MTech

PO CO	PO1	PO2	PO3
CO1	2	1	-
CO2	2	1	-
CO3	2	1	-
CO4	2	1	-
CO5	1	2	-
CO6	2	2	-

CO-PO Mapping - PG ASAS

PO CO	PO1	PO2
CO1	2	1
CO2	2	1
CO3	2	1
CO4	2	1
CO5	1	2
CO6	2	2

Syllabus

Soft Skills

Introduction: Campus to corporate transition based on the expectations of the corporate

Communication and listening skills: Professional & Academic Communication Skills, Interpersonal Communication in Teams, Cross-cultural & Global Communication, Netiquette in Corporate Settings, Communicating in Hybrid & Remote Teams

Assertiveness skills: Necessity of being Assertive, Passive Aggression & Aggressiveness, Assertive & Non-assertive differences, Dealing with demands, criticism & Compliments, Assertive Techniques

Presentation skills: Adult learning principles, Preparation, planning, practice delivery, Principles of Effective Technical Presentations, Verbal & Non-verbal Communication in Presentations, Effective Slide Design, Using AI & Digital Tools for Presentations

Presentations by the students

Perception: Locus of control, Bridging Self & Social perception, Attribution theories, Importance of Perspective Taking, Impression management

Goal setting & Time management: Aligning Personal, Academic & Career Goals, SMART Goals, SWOT Analysis, Time Management Strategies, Prioritization Techniques & Planning Tools, Avoiding Procrastination & Distractions

Verbal

Vocabulary: Familiarize students with the etymology of words, help them realize the relevance of word analysis and enable them to answer vocabulary based MCQ questions through practice in etymology, synonyms, antonyms, commonly confused words, and contextual usage.

Grammar: Train students to understand the nuances of English Grammar and thereby enable them to spot grammatical errors in sentences.

Reasoning: Stress the importance of understanding the relationship between words through analogy questions. Emphasize the importance of avoiding the gap (assumption) in arguments/statements/ communication.

Oral Communication Skills: Improve the professional spoken communication by introducing corporate register, workplace expressions, and structured group activities such as debates and panel discussions.

Aptitude

Numbers - Power Cycles, Divisibility, and HCF & LCM, **Percentages** - Profit, Loss & Discount, and Simple & Compound Interest, **Ratio, Proportion & Variation** - Alligations, Mixtures, and Partnership Problems, **Averages**, **Time and Work** - Pipes & Cistern, and Work Equivalence, **Time, Speed and Distance** - Boats & Streams, Races and Circular tracks, **Statistics** - Mean, Median, Mode, Standard Deviation and Quartile Deviation, **Data Interpretation** - Tables, Bar Diagrams, Line Graphs, Pie Charts, Caselets and Mixed Varieties, **Equations** – Linear and Quadratic Equations, **Logarithms, Inequalities, Puzzles** - Optimization Puzzles, Fermi Problems, Game-Based Puzzles, Sudoku, Anagrams, Futoshiki and Rebus Puzzles, **Speed Mathematics, Shortcuts.**

References

Soft Skills

1. Student's Guide For Career Planning, CIR, 2018
2. Andrew J DuRbin , "Applied Psychology: Individual and organizational effectiveness", Pearson- Merrill Prentice Hall, 2004
3. Michael G Aamodt, "An Applied Approach, 6th edition", Wadsworth Cengage Learning, 2010
4. Robert Bolton, Dorothy Grover Bolton, "People Style at Work..and Beyond: Making Bad Relationships Good and Good", Ridge Associates Inc., 2009
5. John Hayes "Interpersonal skills at work", Routledge, 2003
6. Nord, W. R., Brief, A. P., Atieh, J. M., & Doherty, E. M., "Meanings of occupational work: A collection of essays (pp. 21- 64)", Lexington, MA: Lexington Books, 1990
7. Mark J Martinko, "Attribution theory: an organizational perspective", St. Lucie, 1995
8. Miles Hewstone,"Attribution Theory: Social and Functional Extensions", Blackwell, 1983
9. Stephen Covey, "The habits of highly effective people", Free press Revised edition, 2004
10. Kenneth H Blanchard , "The 25 Best Time Management Tools & Techniques: How to Get More Done Without Driving Yourself Crazy" , Peak Performance Press, 1st edition 2005
11. Kenneth H. Blanchard and Spencer Johnson, "The One Minute Manager", William Morrow, 1984

Verbal

1. Verbal Skills Workbook (PG), CIR
2. Erica Meltzer, "The Ultimate Guide to SAT Grammar"
3. Green, Sharon, and Ira K. Wolf, "Barron's New GRE", Barron's Educational Series, 2011
4. Jeff Kolby, Scott Thornburg & Kathleen Pierce, "Nova's GRE Prep Course"
5. Kaplan, "Kaplan New GRE Premier", 2011-2012
6. Kaplan's GRE Comprehensive Programme
7. Lewis Norman, "Word Power Made Easy", Goyal Publishers, Reprint edition, 1 June 2011
8. Manhattan Prep, "GRE Verbal Strategies Effective Strategies Practice from 99th Percentile Instructors"
9. Pearson- "A Complete Manual for CAT", 2013
10. R.S. Aggarwal, "A Modern Approach to Verbal Reasoning"
11. S. Upendran, "Know Your English", Universities Press (India) Limited, 2015
12. Sharon Weiner Green, Ira K. Wolf, "Barron's New GRE, 19th edition (Barron's GRE)", 2019
13. Wren & Martin, "English Grammar & Composition"
14. www.bbc.co.uk/learningenglish
15. www.cambridgeenglish.org
16. www.englishforeveryone.org
17. www.merriam-webster.com

Aptitude

1. Arun Sharma, "How to Prepare for Quantitative Aptitude for the CAT Common Admission Test", Tata Mc Graw Hills, 5th Edition, 2012
2. Arun Sharma, "How to Prepare for Logical Reasoning for the CAT Common Admission Test", Tata Mc Graw Hills, 2nd Edition, 2014
3. Arun Sharma, "How to Prepare for Data Interpretation for the CAT Common Admission Test", Tata Mc Graw Hills, 3rd Edition, 2015
4. R.S. Aggarwal, "Quantitative Aptitude For Competitive Examinations", S. Chand Publishing, 2015
5. R.S. Aggarwal, "A Modern Approach To Verbal & Non-Verbal Reasoning", S. Chand Publishing, Revised -2015
6. Sarvesh Verma, "Quantitative Aptitude-Quantum CAT", Arihant Publications, 2016
7. www.mbatious.com
8. www.campusgate.co.in
9. www.careerbless.com

Evaluation Pattern

Assessment	Internal	External
Continuous Assessment (CA)* – Soft Skills	30	-
Continuous Assessment (CA)* – Aptitude	10	25
Continuous Assessment (CA)* – Verbal	10	25
Total	50	50
Pass / Fail		

*CA - Can be presentations, speaking activities and tests.

CIR Course – PG – Paper II

Sem: 2	26CDS111 Career Development Skills II	L-T-P-C: 0 0 3 1
---------------	--	-------------------------

Pre-requisite:

Willingness to learn, team spirit, foundational proficiency in English language and communication and knowledge of high school level arithmetic.

Course Objectives:

- To equip students with conflict management strategies for effective teamwork and collaboration.
- To prepare students for recruitment processes through interview etiquette, group discussions and professional conduct.
- Help students to sharpen their problem solving and reasoning skills
- Empower students to communicate effectively by using the correct diction, grammar, comprehension and verbal reasoning skills

Course Outcomes:

CO1: Soft Skills - To demonstrate good interpersonal skills, solve problems and effectively participate in group discussions.

CO2: Soft Skills - To write technical resumes and perform effectively in interviews.

CO3: Aptitude - To identify, investigate and arrive at appropriate strategies to solve questions in geometry, probability, permutations & combinations, and statistics by managing time effectively.

CO4: Aptitude - To investigate, understand and use appropriate techniques to solve questions in logical reasoning and data analysis by managing time effectively.

CO5: Verbal - To be able to use diction that is more refined and appropriate and to be competent in knowledge of grammar to correct and improve sentences

CO6: Verbal - To be able to examine, interpret and investigate passages and to be able to generate ideas, structure them logically and express them in spoken or written style that is comprehensible to the audience/recipient.

CO-PO Mapping - MTech

PO CO	PO1	PO2	PO3
CO1	2	1	-
CO2	2	1	-
CO3	2	1	-
CO4	2	1	-
CO5	1	2	-
CO6	2	2	-

CO-PO Mapping - PG ASAS

PO CO	PO1	PO2
CO1	2	1
CO2	2	1
CO3	2	1
CO4	2	1
CO5	1	2
CO6	2	2

Syllabus

Soft Skills

Corporate Culture and Managing Diversity: Understanding Corporate Culture & Organizational Values, Corporate Culture in Indian & Global Organizations

Unconscious Bias, Adapting to Corporate Culture as a New Professional, Diversity Policies & Best Practices

Critical Thinking & problem solving: Problem Identification & Problem Framing, Problem solving process, Handling Ambiguity & Open-Ended Problems, Brainstorming techniques for Critical Thinking, Agile Thinking & Continuous Improvement

Conflict Management: Intrapersonal, Interpersonal & Task-Based Conflict, Conflict Management Styles, Steps in Conflict Resolution, Conflict in Hybrid & Remote Work Environments, Learning from Conflict

Team building and working effectively in teams: Communication & Collaboration in Teams, Stages of Team Development, Diversity, Inclusion & Cultural Sensitivity in Teams, Trust, Accountability & Psychological Safety, Leadership Dynamics

Group discussion: Critical Thinking & Perspective Building, Clarity, Fluency & Professional Language, Body Language & Eye Contact, Handling Interruptions & Disagreements, Self-Reflection & Improvement

Interview skills: Types of Interviews (Technical, HR, Managerial, Research), Understanding Job Descriptions & Role Expectations, Competency-Based & Behavioural Interviews, STAR Technique for Behavioral Questions, AI-Based Interview Screening & Preparation, Interview Netiquette & Modern Interview Contexts

Mock Interview: One on one Mock interview process for all the students

Verbal

Vocabulary: Enhance vocabulary skills through focused practice in idioms, phrasal verbs, and sentence completion, with emphasis on contextual usage and MCQ-based strategies.

Grammar: Develop grammatical accuracy by training students to identify and correct sentence-level errors through sentence correction and cloze test practice.

Reasoning: Strengthen critical reasoning and logical thinking skills through systematic practice in critical reasoning questions and logical ordering of sentences.

Reading Comprehension: Equip students with effective strategies to analyze and respond to reading comprehension passages through guided discussion and practice.

Public Speaking Skills: Enhance oral communication through storytelling for professional success and personal branding, and assess speaking competence through individual speeches and open forum discussions on current affairs.

Writing Skills / Professional Communication: Develop formal writing competence with emphasis on professional email etiquette and the drafting of job application and workplace emails.

Aptitude

Sequence, Series, Geometry - 2D, 3D, Coordinate Geometry, and Heights & Distance,

Permutations & Combinations – Circular & Linear Arrangements and Derangements, **Probability** - Addition & Multiplication Theorems and Conditional Probability, **Logical Reasoning** - Arrangements, Sequencing, Scheduling, Venn Diagram, Network Diagrams, Binary Logic, and Logical Connectives, Clocks, Calendars, Cubes, Non-Verbal reasoning, Blood Relations, Direction Test, Syllogisms, Series, Odd One Out, Coding & Decoding, Cryptarithmic and Input-Output Reasoning, **Data Sufficiency**, **Campus Recruitment Papers, Puzzles** - Lateral Thinking Problems, Riddles, Logical Puzzles, Deduction Puzzles, Calculation Puzzles, and Other Interview Puzzles, **Time Management Strategies**.

References

Soft Skills

1. Student's Guide for Career Planning, CIR
2. Thomas L. Quick, "Successful team building", AMACOM Div American Mgmt Assn, 1992
3. Brian Cole Miller, "Quick Team-Building Activities for Busy Managers: 50 Exercises That Get Results in Just 15 Minutes", AMACOM; 1 edition, 2003.
4. Patrick Lencioni, "The Five Dysfunctions of a Team: A Leadership Fable", Jossey-Bass, 1st Edition, 2002
5. How to Win Friends & Influence People, Dale Carnegie, 1998 Edition.
6. I'm OK--You're OK, Thomas A. Harris, 2012 Edition.
7. Personality Development and Soft Skills, Barun K.Mitra, Oxford, 2nd Edition
8. Effective Group Discussion, Theory and Practice, Gloria J. Galanes McGraw Hill, 15th Edition

Verbal

1. Verbal Skills Workbook (PG), CIR
2. "GMAT Official Guide" by the Graduate Management Admission Council, 2019
3. Arun Sharma, "How to Prepare for Verbal Ability And Reading Comprehension For CAT"
4. Joern Meissner, "Turbocharge Your GMAT Sentence Correction Study Guide", 2012
5. Kaplan, "Kaplan GMAT 2012 & 13"
6. Kaplan, "New GMAT Premier", Kaplan Publishing, U.K., 2013
7. Manhattan Prep, "Critical Reasoning 6th Edition GMAT"
8. Manhattan Prep, "Sentence Correction 6th Edition GMAT"
9. Mike Barrett "SAT Prep Black Book The Most Effective SAT Strategies Ever Published"
10. Mike Bryon, "Verbal Reasoning Test Workbook Unbeatable Practice for Verbal Ability, English Usage and Interpretation and Judgement Tests"
11. www.bristol.ac.uk/arts/skills/grammar/grammar_tutorial/page_55.htm
12. www.campusgate.co.in

Aptitude

1. Arun Sharma, "How to Prepare for Quantitative Aptitude for the CAT Common Admission Test", Tata Mc Graw Hills, 5th Edition, 2012

2. Arun Sharma, “How to Prepare for Logical Reasoning for the CAT Common Admission Test”, Tata Mc Graw Hills, 2nd Edition , 2014
3. Arun Sharma, “How to Prepare for Data Interpretation for the CAT Common Admission Test”, Tata Mc Graw Hills, 3rd Edition , 2015
4. R.S. Aggarwal, “Quantitative Aptitude For Competitive Examinations”, S. Chand Publishing , 2015
5. R.S. Aggarwal, “A Modern Approach To Verbal & Non-Verbal Reasoning”, S. Chand Publishing , Revised -2015
6. Sarvesh Verma, “Quantitative Aptitude-Quantum CAT” , Arihant Publications , 2016
7. www.mbatious.com
8. www.campusgate.co.in
9. www.careerbless.com

Evaluation Pattern

Assessment	Internal	External
Continuous Assessment (CA)* – Soft Skills	30	-
Continuous Assessment (CA)* – Aptitude	10	25
Continuous Assessment (CA)* – Verbal	10	25
Total	50	50

*CA - Can be **presentations, speaking activities and tests.**