M. TECH – COMPUTER SCIENCE AND ENGINEERING
Department of Computer Science and Engineering

The M.Tech. Computer Science and Engineering program is offered at Amrita Vishwa Vidyapeetham by the Department of Computer Science and Engineering in the Amrita School of Engineering.

The field of Computer Science and Engineering is a constantly evolving one and drives the technological trends in today's world. The foundations of these technological trends are rooted in the core concepts and principles of the field of Computer Science. This master's programme is designed to produce graduates who can apply fundamental knowledge of mathematics, programming, problems solving and computing systems to model and solve problems in the real-world and provide a strong foundation to be able to adapt to emerging technological trends that are increasingly end-to-end systems driven combining both software and hardware.

With a view towards developing skilled and adaptable graduate students in Computer Science and Engineering the program curriculum has been framed to incorporate and deliver on foundational aspects of mathematics and computer science, programming and problem solving, system and network design and upcoming key technologies and tools for solving real-world problems. The courses include core courses in Mathematics and Computer Science and a bouquet of soft-core courses spanning the core foundations of programming and problem solving, software engineering, computing systems and key technological trends like Cloud Computing, IoT, AI, and Data Science. There are also wide ranges of electives in areas spanning Internet of Things, Fog, Edge and Cloud Computing, Cyber Security, Web Science, Mobile Computing, Full Stack Engineering etc. At the end of the course, the student would have developed strong foundational skills and strength in selected key technologies so that they can take up advanced research as part of the thesis component and become professionals in this area.

The degree is suitable for students with a bachelor's degree in a computing related field as well as students who want to demonstrate computer science expertise in addition to a degree in another field. The curriculum has been designed to prepare students for a broad range of rewarding careers like Software Developer, Software Architect, Full Stack Developer, Technical Architect, AI Engineer, Big Data Engineer, Application analyst, Data Scientist, Computer Network Architect, Research Analyst, and Senior Research Engineer etc. As a part of the programme during the period of study, students have the opportunity to intern at leading companies and R&D labs for a period of upto to one year. There are opportunities for the students to take up a semester or one year study at International Universities like Vrije University, Netherlands, UC Davis, UNM for an exchange programme or to pursue a dual degree programme.

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 inorder 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 relates to case studies, lab implementations and dissertation.
3. PO4 and PO5 (adopted from NBA) relates to dissertation as well as to case studies.
4. PO6 explicitly mentions about 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 to our courses.

M.Tech Computer Science & Engineering

CURRICULUM

I Semester

Course Code	Type	Subject	L T P	Credits
24CS601	FC	Advanced-Data Structures and Algorithms	3-0-2	4
24MA602	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
24RM604	FC	Research Methodology	2-0-0	2
23AVP601	HU	Amrita Values Program*		P/F
22AVP103	HU	Mastery Over Mind	1-0-2	2
23HU601	HU	Career Competency 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	3-0-0	3
23HU611	HU	Career Competency II	0 0 3	1
		Credits		**19**

III Semester

Course Code	Type	Subject	L T P	Credits
	E	Elective 3	2-0-2	3
	E	Elective 4	3-0-0	3
24CS798	P	Dissertation-I		10
		Credits		**16**

IV Semester

Course Code	Type	Subject	L T P	Credits
24CS799	P	Dissertation-II		16

Total Credits 71

List of Courses

FOUNDATION CORE

Course Code	Subject	L T P	Credits
24CS601	Advanced Data Structures and Algorithms	3-0-2	4
24MA602	Mathematical Foundations for Computer Science & Engineering	3-1-0	4
24RM604	Research Methodology	2-0-0	2

SOFT CORE

Course Code	Subject	L T P	Credits
24CS631	Software Engineering with Agile and DevOps	3-0-2	4
24CS632	Distributed Systems	3-0-2	4
24CS633	Advanced Networks	3-0-2	4
24CS634	Foundations of Cyber Security	3-0-2	4
24CS635	Full Stack Development	3-0-2	4
24CS636	Modern Database Management Systems	3-0-2	4
24CS637	Advanced Operating Systems	3-0-2	4
24CS638	Machine Learning	3-0-2	4
24CS639	Foundations of AI	3-0-2	4

OPEN ELECTIVES

The students have the option of choosing electives from other M.Tech programmes like Artificial intelligence, Data Science, Cyber-Security etc.

ELECTIVES

Course Code	Subject	L T P	Credits
24CS731	Quantum Computing	2-0-2	3
24CS732	Design Patterns	3-0-0	3
24CS733	Mobile Application Development	2-0-2	3
24CS734	Software Quality Assurance and Testing	2-0-2	3
24CS735	Fog and Edge Computing	2-0-2	3
24CS736	Augmented Reality and Virtual Reality	2-0-2	3
24CS737	Block Chain Technology	2-0-2	3
24CS738	Cloud Computing and Services	2-0-2	3
24CS739	Deep Learning	2-0-2	3
24CS740	Image and Video Processing	2-0-2	3
24CS741	Internet of Things	2-0-2	3
24CS742	Offensive Cyber Security	2-0-2	3
24CS743	Project and Finance Management	3-0-0	3
24CS744	Wireless and Mobile Networks	2–0-2	3
24CS745	Software Defined Systems	2-0-2	3
24CS746	Secure Coding	2-0-2	3
24CS747	Robotics and Control	2-0-2	3
24CS748	Evolutionary Robotics	2-0-2	3
24CS749	Knowledge Networks	3-0-0	3
24CS750	Foundations of Data Science	2-0-2	3
24CS751	Big Data Analytics	3-0-0	3
24CS752	High Performance Computing	2-0-2	3
24CS753	Medical Imaging	3-0-0	3

SYLLABUS

FOUNDATION CORE

24CS601	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, Amortized Analysis. Array based structures, lists and. Advanced Data Structures - Dictionaries, hash tables, bloom filters, binary search trees, interval and range trees; skip lists.

Foundations and Applications of Divide-and-Conquer, Greedy techniques, Dynamic Programming, Backtracking and Branch and Bound. Applications of graph algorithms: Topological sort, Strongly Connected Components, Bi-connected Components, Bridges, Articulation points. All Pair Shortest Paths, Single Source Shortest Paths.

Flow Networks: Ford-Fulkerson, Edmonds Karp, Applications of maximum flows - Efficient algorithms for maximum bipartite matching. NP-Completeness: Important NP-Complete Problems, Polynomial time reductions, Approximation algorithms, Parallel Algorithms (overview): Tree Contraction - Divide and Conquer - Maximal Independent Set.

Text Books/References

1. Goodrich M T, Tamassia R and Michael H. Goldwasser, "Data Structures and Algorithms in Python++", Wiley publication, 2013.
2. Cormen T H, Leiserson C E, Rivest R L and Stein C. Introduction to Algorithms, Prentice Hall of India Private Limited, Third Edition; 2009.
3. Michael T Goodrich and Roberto Tamassia, "Algorithm Design and Applications", John Wiley and Sons, 2014.
4. Motwani R, Raghavan P. Randomized algorithms. Cambridge university press; 1995.
5. Udi Manber, "Algorithms : A Creative Approach", Pearson, First Edition, 1989.
6. Vijay V. Vazirani. Approximation Algorithm, Springer; 2003.
7. Steven S.Skiena, "The Algorithm Design Manual", Springer, Third Edition, 2020.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours.

24MA602	MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE & ENGINEERING	3-1-0-4

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 variable 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, Basis and Dimensions - Change of Basis, Orthogonality and Gram Schmidt Process - Four Fundamental Spaces: Column Space, Null Space, Row Space and Left Null Space

- Projection, Least Squares and Linear Regression - Eigen Value Decomposition and Diagonalization - Special Matrices, Similarity and Algorithms - Singular Value Decomposition.

Probability models and axioms, Bayes' rule, Conditional Probability, Independence - Discrete random variables: probability mass functions(PMF), expectations, multiple discrete random variables: joint PMFs - Continuous random variables: probability density functions (PDF), expectations, multiple continuous random variables, continuous Bayes rule - Binomial, Poisson, Geometric, Exponential, Uniform and Normal Distributions - Derived distributions; convolution; covariance and correlation - Weak law of large numbers, central limit theorem.

Parameter Estimation - Hypothesis Testing - Application of Hypothesis Testing in Statistics: case studies- Regression - Analysis of Variance - Non parametric Hypothesis Tests - Experiment Design

Text Books/References

1. Gilbert Strang, Linear algebra for everyone, Wellesley-Cambridge Press, 2020.
2. Axler Sheldon, Linear algebra done right, Springer Nature, 2024.
3. Howard Anton and Chris Rorrers," Elementary Linear Algebra", Tenth Edition, 2010 John Wiley & Sons, Inc.
4. David Forsyth, "Probability and Statistics for Computer Science", Springer international publishing, 2018
5. Ernest Davis, "Linear Algebra and Probability for Computer Science Applications", CRC Press, 2012

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours.

24RM604	RESEARCH METHODOLOGY	2-0-0-2

Pre-Requisite(s):
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. Bordens, K. S. and Abbott, B. B., "Research Design and Methods – A Process Approach", 11thEdition, McGraw-Hill, 2022.
2. Roy Sabo and Edward Boone, "Statistical Research Methods: A Guide for Non-Statisticians", Springer, 2013.
3. Davis, M., Davis K., and Dunagan M., "Scientific Papers and Presentations", 3rd Edition, Elsevier Inc., 2013.
4. Ron Iphofen (Ed), "Handbook of Research Ethics and Scientific Integrity", Springer, 2020.
5. Elsevier, "Ethics in Research & Publication", https://www.elsevier.com/____data/assets/pdf_file/0008/653885/Ethics-in-research-and-publication-brochure.pdf

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Case Study)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours.

24CS798	DISSERTATION- I	10

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 the research problem 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

24CS799	DISSERTATION- II	16

Pre-Requisite(s): Dissertation Phase I
Course Type:

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 demonstrate the effectiveness of the proposed solution.
- The student is also expected to effectively communicate the scholarly outcomes as presentations, report, and publish the same 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

SOFT CORE

24CS631	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: Compare and contrast the differences between Agile and other project management methodologies

CO2: Interpret and apply various principles, phases and activities of the Scrum methodology.

CO3: Apply Agile Testing principles for real life situations and understand the basics of SAFe for scaled agile

CO4: Identify and apply various tools for Agile development and CI/CD

CO5: Implement DevOps principles for CI/CD

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

Introduction to Agile: Agile versus traditional method comparisons and process tailoring Software Process Models – overview, Various Agile methodologies - Scrum, XP, Lean, and Kanban, Agile Manifesto, Scrum and artifacts, Agile Requirements - User personas, story mapping, user stories, 3Cs, INVEST, acceptance criteria, sprints, requirements, product backlog and backlog grooming; Tools: Agile tracking tools; Definition of Done, Definition of Ready;

Estimation; Agile forecasting and project Management - Big visible information radiators, velocity, progress tracking, Track Done pattern, project forecasting, Ux Design, Control the Flow, Sprint Planning, Create product roadmap Sprints: Iterations/Sprints Overview. Velocity Determination, Iteration Planning Meeting, Iteration, Planning Guidelines, Development, Testing, Daily Stand-up Meetings, Progress Tracking, Velocity Tracking, Monitoring and Controlling.

Scaled agile frameworks: SAFe, Scrum@Scale, Disciplined Agile Testing: Functionality Testing, UI Testing, Performance Testing, Security Testing, Tools - Selenium Agile Testing: Principles of agile testers; The agile testing quadrants, Agile automation, Test automation pyramid. Test automation using UI test tools such as Selenium, Writing unit tests; Test Driven Development.

DevOps: Continuous Integration and Continuous Delivery CI/CD: Jenkins Creating pipelines, Setting up runners Containers and container orchestration (Dockers and Kubernetes) for application development and deployment; Checking build status; Kubernetes, Run a container image within a kubernetes cluster Fully Automated Deployment; Continuous monitoring with Nagios; Introduction to DevOps on Cloud.

Text Books/References

1. Learning Agile: Understanding Scrum, XP, Lean, and Kanban, By Andrew Stellman, Jennifer Greene, 2015, O Reilly.
2. Rubin KS. “Essential Scrum: a practical guide to the most popular agile process”. Addison-Wesley; 2012.
3. Cohn M. “User stories applied: For agile software development”. Addison-Wesley Professional; 2004.
4. Crowder JA, Friess S. “Agile project management: managing for success”. Cham: Springer International Publishing; 2015.
5. Agile Testing: A Practical Guide For Testers And Agile Teams, Lisa Crispin, Janet Gregory, Pearson, 2010.
6. More Agile Testing: Learning Journeys for the Whole Team By Janet Gregory, Lisa Crispin, Addison Wesley, 2015.
7. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive ... By Sricharan Vadapalli, Packt, 2018.
8. DevOps: Puppet, Docker, and Kubernetes By Thomas Uphill, John Arundel, Neependra Khare, Hideto Saito, Hui-Chuan Chloe Lee, Ke-Jou Carol Hsu, Packt, 2017.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS632	DISTRIBUTED SYSTEMS	3-0-2-4

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 and types of distributed systems – Introduction to P2P systems, Edge Networks, CPS, etc, architecture of DS - overview of processes - A Taxonomy of Distributed Systems, scalable performance, load balancing, and availability. Models of computation - shared memory and message passing system— synchronous and asynchronous systems. Communication in Distributed Systems - Remote Procedure Calls and Message Oriented Communications and implementation, High-level communication and publish-subscribe in Mapreduce

Logical time and event ordering, Global state and snapshot algorithms, clock synchronization, distributed mutual exclusion, leader election, deadlock detection, termination detection, Distributed Databases, implementations over a simple distributed system and case studies of distributed databases and systems - Distributed file systems: Examples from Dropbox, Google FS (GFS)/ Hadoop Distributed FS (HDFS), Bigtable/HBase MapReduce, RDD

Consistency control: Data Centric Consistency, Client Centric Consistency, Replica Management, Fault tolerance and recovery: basic concepts, fault models. Case Studies from Apache Spark, Edge Networks, Cyber-Physical Systems, Google Spanner, Amazon Aurora, BlockChain Systems etc.

Tools: MPI, OpenMP

Text Books/References

1. Andrew S. Tannenbaum and Maarten van Steen, Distributed Systems: Principles and Paradigms, Third Edition, Prentice Hall, 2017.
2. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2012.
3. Garg VK. Elements of distributed computing. John Wiley & Sons; 2002.
4. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Distributed Systems: Concepts and Design, Fifth Edition, Pearson Education, 2017.
5. Fokkink W. Distributed algorithms: an intuitive approach. Second Edition, MIT Press; 2018.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS633	ADVANCED NETWORKS	3-0-2-4

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 with 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 with 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, Architectural principle, Layering, Names and addresses. TCP/IP suite of protocols, TCP extensions for high-speed networks, multimedia networking applications, IPv6: API for IPv6. QoS in IP Network, traffic engineering and analysis, SNMP and access control.

Recent trends in Wireless communication networks, wireless sensor networks, mobile ad-hoc networks, mobility based protocols, information aggregation, information storage and query, localization services.

SDN: Data and Control Plane, Open flow Control, Network Function Virtualization for IoT. 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. James F. Kurose & Keith W. Ross, Computer Networking: A Top-Down Approach, 7/E. Pearson Education India, 2017.
2. Douglas E Comer, Computer networks and Internets, 6th Edition, Pearson Education, 2015
3. Goransson, P., Black, C., & Culver, T. Software defined networks: a comprehensive approach. Morgan Kaufmann,2016.
4. Online Resources: Technical papers in course related topics and IEEE Standards documents

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS634	FOUNDATIONS OF CYBERSECURITY	3-0-2-4

Pre-Requisite(s): None
Course Type: Project

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, Cryptography, Confusion vs Diffusion, Stream vs Block ciphers, Private-Key vs Public Key Cryptography, Cryptanalysis. Feistal Networks and Non Feistal Networks, Key Exchange, Public Key Encryption systems. Hash Functions, Message Digest, Signing and Verification, X509 certificates, SSL, TLS, HTTPS, Open SSL.

Authentication and Authorization: Identity and Access Management, Factors, Multi-factor, Kerberos, Role based Access Control. Operating System Security: Windows and Linux,

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, Security issues in Big Data and Cloud. Implementing Cryptographic schemes on datasets. Tools: Snort, NPM, Open VAS, Burp Suite, Nessus. Case Study: Implementing Cryptographic schemes on datasets. Understanding vulnerabilities using security tools: Snort, NPM, Open VAS, Burp Suite, Nessus.

Text Books/References

1. William Stallings, Cryptography and Network Security Principles and Practice, 8th Edition, 2023.
2. Jose Manuel Ortega, Mastering Python for Networking and Security, Packt Publishing, 2018
3. R. Sarma Danturthi, Database and Application Security: A Practitioner's Guide, Addison-Wesley Professional; 1st edition, 2024
4. Padmanabhan TR, Shyamala C K, and Harini N, “Cryptography and Security”, First Edition, Wiley India Publications, 2011
5. Jose Manuel Ortega, Mastering Python for Networking and Security, Packt Publishing, 2018.
6. Matt Bishop, Computer security: Art and Science, Vol. 2, Addison-Wesley, 2012.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS635	FULL STACK DEVELOPMENT	3-0-2-4

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, containerization with Docker, and deploy applications using hosting services like Heroku and AWS, alongside 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- DOM Manipulation and Events- 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.

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 Set Book by Jon Duckett Professional
2. Full-Stack JavaScript Development by Eric Bush.
3. Mastering Full Stack React Web Development Paperback – April 28, 2017 by Tomasz Dyl,Kamil Przeorski, Maciej Czarnecki
4. JavaScript for Web Developers Book by Nicholas C. Zakas
5. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic
6. Websites by Robin Nixon Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB. Copyright © 2015 BY AZAT MARDAN

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS636	MODERN DATABASE MANAGEMENT SYSTEMS	3-0-2-4

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

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	2			
CO2	3	2	1	1		1
CO3	3	2	1	1		1
CO4	3	2	1	1		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.

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.

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. Ramesh Elmasri and Shamkant B Navathe, "Fundamentals of Database Systems", Fifth Edition, Pearson Education India, 2008.
2. Silberschatz A, Korth H F and Sudharshan S, "Database System Concepts", Sixth Edition, Tata McGraw-Hill Publishing Company Limited, 2010.
3. Niall O'Higgins, "MongoDB and Python", O'reilly, 2011.
4. Hector Garcia-Molina, Jeff Ullman and Jennifer Widom, "Database Systems: The Complete Book", Pearson, 2011.
5. Raghu Ramakrishnan and Johannas Gehrke, "Database Management Systems", Third Edition, McGraw-Hill, 2003.
6. Andreas Meier, Michael Kaufmann, "SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management", Springer Verlag 2019.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS637	ADVANCED OPERATING SYSTEMS	3-0-2-4

Pre-Requisite(s): None
Course Type: Lab

Course Objectives

- To gain knowledge in advanced topics of operating system design and implementation
- To learn about operating system structuring, synchronization, communication, and scheduling in parallel and concurrent systems
- To understand communication mechanisms in distributed systems, virtualization, file, and disk structure management

Course Outcomes

CO1: Analyze and apply synchronization principles in parallel processing and distributed systems

CO2: Describe and analyze the memory management and its allocation policies in cluster machines

CO3: Understand Virtual Machines and its interaction with a Hypervisor through practical implications

CO4: Evaluate the storage management policies with respect to different storage management technologies

CO-PO Mapping

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

Syllabus

Concurrent Execution: Threads, event systems, asynchronous/synchronous I/O, Parallelism, Ordering, and Races, Dynamic Data Race Detector for Multi-Threaded Programs, Discussions of synchronization with an emphasis on monitors, On Optimistic Methods for Concurrency Control, Concurrency Control and Recovery, Communication using lightweight remote procedure call (RPC)

Memory Management: virtual memory, NUMA machines, memory allocators – Hoard Scalable Memory Allocator, Memory Resource Management in VMware, Global Memory Management in Cluster machines, Virtualization: Machine virtualization, binary instrumentation, VMware design etc.

File Systems and Disk: File system interfaces, Networked file systems, AFS, The Design and Implementation of a Log-Structured File System, File system extensibility, non-disk file systems, A Case for Redundant Arrays of Inexpensive Disks (RAID), Using Model Checking to Find Serious File System Errors Big Data System. Case studies on Mobile operating systems, Android, iOS, Samsung etc

Text Books/References

1. Mukesh Singhal, Niranjan Shivaratri, Advanced Concepts in Operating Systems: Distributed, Database, and Multiprocessor Operating Systems, McGraw Hill, 2017.
2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson, 2022.
3. Max Hailperin, Operating Systems and Middleware: Supporting Controlled Interaction. Creative Media Partners LLC, 2023.
4. Ajit Singh, Operating System: Simply in Depth, Amazon Digital Services LLC, 2020.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS638	MACHINE LEARNING	3-0-2-4

Pre-Requisite(s): Basics of Linear Algebra, Probability Theory and Optimization: Vectors, Inner product, Outer product, Inverse of a matrix, Eigenanalysis, 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: Lab

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 basic of ML, learning paradigms and concepts of regression

CO2: Design and develop classifier models and evaluate their performance

CO3: Acquire skills to build probabilistic model and deep network models for classification

CO4: Develop and build clustering models for real world applications

CO5: Understand and apply the concepts of dimensionality reduction and Reinforcement Learning

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 - different forms of learning- Linear regression - Ridge and Lasso regression, Logistic regression, Discriminant Functions and models, Bayesian regression, regression with basic functions.

Classification - Perceptron –Multilayer Perceptron - Feed forward network - Backpropagation – Support vector machine - Decision trees - evaluation of classifiers – bias and variance. Gaussian mixture models -- Expectation-Maximization - Naive Bayes classifier - Ensemble Methods - Bagging – Boosting -Time series Prediction and Markov Process - Introduction to deep learning - Convolutional neural networks - application of classification algorithm

Clustering - K-means – Hierarchical and Density Based Clustering – DBSCAN- Assessing Quality of Clustering - Dimensionality reduction - Principal Component Analysis - Introduction to Reinforcement Learning.

Text Books/References

1. Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020
2. Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.
3. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
4. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997
5. Duda, Richard O., and Peter E. Hart. Pattern classification. John Wiley & Sons, 2006.
6. Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining concepts and techniques third edition." The Morgan Kaufmann Series in Data Management Systems 5.4 (2011): 83-124.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS639	FOUNDATIONS OF AI	3-0-2-4

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. Russell, Stuart Jonathan, Norvig, Peter, Davis, Ernest. Artificial Intelligence: A Modern Approach. United Kingdom: Pearson, 2010.
2. Deepak Khemani. A First Course in Artificial Intelligence. McGraw Hill Education (India), 2013.
3. Denis Rothman. Artificial Intelligence by Example, Packt, 2018.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

ELECTIVES

24CS731	QUANTUM COMPUTING	2-0-2-3

Pre-Requisite(s): Probability
Course Type: Lab

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 computation with Qubits

CO2: Apply Quantum algorithms -Fourier Transform and Grovers amplification

CO3: Apply Quantum problem solving using tree search

CO4: Understand and explore the models of Quantum Computer and Quantum Simulation tools

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

Introduction to quantum physics - -ary Evolution - Quantum Mechanics - Hilbert space - Quantum Time Evolution -Von Neumann Entropy - Measurement – Schrodinger Equation - Heisenbergs uncertainty principle - Randomness - Computation with Qubits -Matrix Representation of Serial and Parallel Operations - Quantum Boolean Circuits -Periodicity - Quantum Fourier Transform - N-ary Transforms - Search and Quantum Oracle - Grovers Amplification - Circuit Representation - Speeding up the Traveling Salesman Problem -The Generate-and -Test Method - Quantum Problem - Solving -Heuristic Search - Quantum Tree Search -Tarratacas Quantum Production System.

Problem Solving-Rules-Logic-based operators-Frames - Categorial representation - Binary

vector representation-Production System-Deduction systems - Reaction systems - Conflict resolution - Human problem – solving - Information and measurement - Reversible Computation-Reversible circuits - Toffoligate – Gate based Quantum Computer – standard gates and their operations.

A General Model of a Quantum Computer - Cognitive architecture - Representation -Quantum Cognition - Decision making - Unpacking Effects - Quantum walk on a graph -Quantum annealing - Optimization problems - Quantum Neural Computation - Applications on Quantum annealing Computer – Development libraries - Quantum Computer simulation toolkits.

Text Books/References

1. Jack D. Hidary, Quantum Computing: An Applied Approach, Firstedition, Springer International Publishing, 2019
2. N. David Mermin, Quantum Computer Science: An Introduction Firstedition, Cambridge University Press, 2007
3. I. Chuang and M.Nielsen, Quantum Computation and Quantum Information, Cambridge University Press, 2012
4. Michael A. Nielsen & Isaac L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition, Cambridge Press.
5. Phillip Kaye, Raymond Laflamme, and Michele Mosca, An Introduction to Quantum Computing, Oxford University Press, 2007.
6. Stephen Barnett, Quantum Information, Oxford University Press, 2009.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS732	DESIGN PATTERNS	3-0-0-3

Pre-Requisite(s): Object oriented programming, UML
Course Type: Theory

Course Objectives

- Understand the common software design problems
- How to use design patterns to solve these problem
- Ability to use the right pattern for a given scenario.

Course Outcomes

CO1: Understand the common software design problems seen in the development process

CO2: Demonstrate the use of various design patterns to tackle these common problems **CO3:**

Identify the most suitable design pattern to address a given software design problem **CO4:**

Analyze existing code for anti-patterns and refactor the code

CO5: Apply best practices of design principles for software design and development

CO-PO Mapping

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

Syllabus

Overview of object-oriented concepts, UML, SOLID, DRY, and YAGNI design principles. Introduction to Patterns, Pattern Categories, Related patterns and Anti-Patterns, Patterns and software architecture, Introduction to Design Patterns: Evolution of design patterns, Description, Taxonomy of Design Patterns, Catalog of design pattern, problem solving by design patterns, Guidelines for selecting and using design pattern.

Creational Patterns: Singleton Pattern, Factory Method Pattern, Abstract Factory Pattern, Builder Pattern, Prototype Pattern. Structural Patterns: Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Façade Pattern, Flyweight Pattern, Proxy Pattern.

Behavioral Patterns: Chain of Responsibility Pattern, Command Pattern, Interpreter Pattern, Iterator Pattern, Mediator Pattern, Memento Pattern, Observer Pattern, State Pattern, Strategy Pattern, Template Method Pattern, Visitor Pattern.

Emerging Trends: Benefits of Pattern in software development, Microservices patterns, Cloud-native patterns, Patterns in Modern Frameworks Spring, .NET, Django.

Text Books/References

1. Gamma & Helm et al, "Design Patterns", Addison Wesley 1999.
2. Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra "Head First Design Patterns", O'Reilly Media, Inc., 2004
3. Frank Bushmann et al, "Pattern oriented Software Architecture", John Wiley & sons, 2001

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS733	MOBILE APPLICATION DEVELOPMENT	2-0-2-3

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

Course Objectives

- To provide a comprehensive understanding of mobile application development for both Android (using Kotlin) and cross-platform (using Flutter/Dart)
- To develop skills in designing intuitive user interfaces, managing data, and integrating external services and APIs
- To understand and apply best practices in mobile app architecture, storage optimization, and user experience design for cross-platforms
- To gain practical experience in integrating cloud services and preparing apps for deployment

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	3	3	2	3	1
CO2	3	3	3	3	3	1
CO3	3	3	3	3	3	2
CO4	3	3	3	3	3	2

Syllabus

Fundamentals for Mobile App Development: Introduction to mobile app development ecosystems - Android architecture and Flutter framework overview - Setting up development environments (Android Studio, VS Code) - Kotlin essentials for Android development - Kotlin syntax, null safety, and functional programming concepts - Dart essentials for Flutter development - Dart syntax and Flutter-specific language features - Creating a basic app structure
- 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 app structure - Android: Intents, Navigation component - Flutter: Navigator, Routes

Data Management and Device Integration: Local data persistence - Android: Shared Preferences, Room database - Flutter: SQLite - Networking and API integration - Android: Retrofit, Moshi - Flutter: http package, JSON serialization - Firebase integration and cloud services - Setting up Firebase for Android and Flutter projects - Firebase Authentication - Cloud Firestore for real-time database - Firebase Cloud Storage - Firebase Cloud Messaging for push notifications - State management - Android: ViewModel, LiveData - Flutter: Provider, Riverpod, or BLoC - Background processing - Android: WorkManager, Services - Flutter: Isolates, background execution - Accessing device features - Camera, location, and sensors integration in both platforms - Notifications and push services - Android: Notification API - Flutter: flutter_local_notifications, Firebase Cloud Messaging

Advanced Topics and App Publishing: Advanced UI and Animations - Android: MotionLayout, Transitions - Flutter: Animation Controller, custom animations - Custom Views/Widgets - Android: Custom Views - Flutter: Custom Widgets, Custom Painter - Testing and Debugging - Android: JUnit, Espresso - Flutter: Unit tests, Widget tests, Integration tests - Performance optimization - Profiling and optimizing apps in both platforms - App architecture patterns - MVVM, Repository pattern, Clean Architecture - Dependency Injection - Android: Hilt - Flutter: get_it, injectable - Preparing for app store submission - App signing, versioning, and distribution for both platforms

Text Books/References

1. How to Build Android Apps with Kotlin - Second Edition , by Alex Forrester, Eran Boudjnah, Alexandru Dumbravan, Jomar Tigcal, May 2023, Publisher(s): Packt Publishing, ISBN: 9781837634934
2. Neil Smyth, "Android Studio 4.2 Development Essentials - Kotlin Edition", Payload Media, 2021
3. Beginning Flutter: A Hands On Guide to App Development , Marco L. Napoli , ISBN: 978-1-119-55082-2, October 2019
4. Marco L. Napoli, "Flutter Complete Reference: Create Beautiful, Fast and Native Apps for any Device", Independently published, 2022
5. Programming Android with Kotlin, by Pierre-Olivier Laurence, Amanda Hinchman -Dominguez, Mike Dunn, G. Blake Meike, December 2021, Publisher(s): O'Reilly Media, Inc., ISBN: 9781492063001

6. Pro Android with Kotlin: Developing Modern Mobile Apps with Kotlin and Jetpack, by Peter Späth, December 2022, Publisher(s): Apress, ISBN: 978-1484287453
7. Flutter Complete Reference 2.0: The ultimate reference for Dart and Flutter, by Alberto Miola
8. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st ed. Edition, by Rap Payne, 2019, Publisher(s): Apress, ISBN: 978-1484251805
9. Android Developer Documentation: https://developer.android.com/docs
10. Flutter Documentation: https://flutter.dev/docs
11. Kotlin Documentation: https://kotlinlang.org/docs/home.html
12. Firebase Documentation: https://firebase.google.com/docs

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS734	SOFTWARE QUALITY ASSURANCE AND TESTING	2-0-2-3

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

Course Objectives

- To identify defects in software by applying the concepts of testing.
- To learn to assess the quality of a software

Course Outcomes

CO1: Understand the basic concepts of Software Quality and standards

CO2: Apply appropriate defect prevention techniques and software quality assurance metrics

CO3: Apply techniques of quality assurance for a given application

CO4: Perform functional and non-functional tests in the life cycle of the software product.

CO5: To build design concepts for system testing and execution

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 - People‘s Quality Expectations, Frameworks and ISO-9126, McCall‘s Quality Factors and Criteria – Relationship. Quality Metrics. Quality Characteristics ISO 9000:2000 Software Quality Standard. Maturity models- Test Process Improvement, Testing Maturity Model. Quality Assurance - Root Cause Analysis, modeling, technologies, standards and methodologies for defect prevention. Fault Tolerance and Failure Containment - Safety Assurance and Damage Control, Hazard analysis using fault-trees and event-trees. Comparing Quality Assurance Techniques and Activities. QA Monitoring and Measurement, Risk Identification for Quantifiable Quality Improvement. Case Study: FSM-Based Testing of Web-Based Applications, Role of AI in SQA.

Quality Revolution, Verification and Validation, Failure, Error, Fault, and Defect, Objectives of Testing, Testing Activities, Test Case Selection White-Box and Black ,test Planning and design, Test Tools and Automation, . Power of Test. Test Team Organization and Management-Test Groups, Software Quality Assurance Group ,System Test Team Hierarchy, Team Building.

System Testing - System Integration Techniques-Incremental, Top Down Bottom Up Sandwich and Big Bang, Software and Hardware Integration, Hardware Design Verification Tests, Hardware and Software Compatibility Matrix Test Plan for System Integration. Built- in Testing. functional testing - Testing a Function in Context. Boundary Value Analysis, Decision Tables. acceptance testing - Selection of Acceptance Criteria, Acceptance Test Plan, Test Execution Test. software reliability - Fault and Failure, Factors Influencing Software, Reliability Models, UI/Ux based testing models.

Text Books/References

1. Software Testing and Quality Assurance-Theory and Practice, Kshirasagar Nak Priyadarshi Tripathy, John Wiley & Sons Inc, 2008
2. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
3. Software Quality Assurance - From Theory to Implementation, Daniel Galin, Pearson Education Ltd UK, 2004.
4. Software Quality Assurance, Milind Limaye, TMH, New Delhi, 2011

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS735	FOG AND EDGE COMPUTING	2-0-2-3

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

Course Objectives

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

Course Outcomes

CO1: Understand the foundations of fog and edge computing networks and different architectures

CO2: Design fog and edge computing based systems and applications using reference architectures

CO3: Understand and apply data collection, analysis, decision making and learning methodologies over the edge for different applications

CO4: Apply optimization techniques for edge and fog computing

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: Introduction to IoT, Fog and Edge Computing, hierarchy of Fog and Edge Computing, edge network, Edge computing architectures, OpenFog Reference Architecture for Fog Computing, Optimization in Fog and Edge Computing, Case Study: open source platforms like Apache Edgent.

Middleware: Middleware for Fog and Edge Computing: Design Issues, Lightweight Container Middleware for Edge Cloud Architectures, Data Management in Fog Computing, Predictive

Analysis to Support Fog Application Deployment, Using Machine Learning for Protecting the Security and Privacy of Internet of Things (IoT) Systems

Applications: Applications of Fog Computing in Big Data Analytics, health monitoring, smart surveillance, smart transportation, Modeling. Simulation of Fog and Edge Computing Environments Using open source platforms like iFogSim Toolkit

Text Books/References

1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019
2. Javid Taheri, Shuiguang Deng, "Edge Computing: Models, technologies and applications", IET, 2020
3. Khaldoun Al Agha, Pauline Loygue, Guy Pujolle, " Edge Networking",Wiley-ISTE,2022.
4. Xin Sun and Amin Vahdat, "Edge Computing: A Primer", CRC Press, 2019.
5. "OpenFog Reference Architecture for Fog Computing", Industry IoT Consortium, OpenFog_Reference_Architecture_2_09_17.pdf (iiconsortium.org
6. "IEEE Standard for Adoption of Openfog Reference Architecture for Fog Computing," Aug. 2018, standard No. 1934-2018", [online] Available: https://standards.ieee.org/standard/1934-2018.html.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS736	AUGMENTED REALITY AND VIRTUAL REALITY	2-0-2-3

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 and design principles of augmented reality and virtual reality.

CO2: Understand the mathematical foundations, hardware, and software development tools required for VR and AR.

CO3: Design and develop immersive AR/VR experiences by applying principles of user interface design, interaction design, and user experience

CO4: Analyze existing code for anti-patterns and refactor the code

CO5: Analyze human factor issues, user performance, sensorial conflict aspects of VR/AR/ MR

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	3	2	3	3	3
CO5	3	2	2	1	1	

Syllabus

Introduction: Introduction to Virtual Reality (VR)-Augmented Reality (AR)-Mixed Reality (MR), Taxonomy, technology and features of AR, the difference between AR, VR, and MR,

challenges with AR, AR systems, and functionality. The geometry of virtual worlds and the physiology of human vision: geometric models, changing position and orientation, axis-angle representations of rotation, viewing transformations, chaining the transformations, fundamentals of the human visual system, depth cues, stereopsis.

Virtual Reality: Introduction, Input and output devices, VR hardware: VR headset, VR controller, and VR glove, Software development tools in VR, User interface (UI) and user experience (UX) design: Spatial Design, Interaction Design, Visual Design, and Feedback and Responsiveness. Augmented Reality: Introduction, Types of AR systems: marker-based, markerless, and location-based AR, AR hardware: AR glass, wearable devices, and smartphone, Software development tools in AR, Principles of AR Design: contextual relevance, spatial awareness, user-centered interaction, and integration of physical and digital elements.

Advanced topics in AR/VR: Augmented Reality and Artificial Intelligence, computer vision in AR/VR, Internet of Things (IoT) integrated AR/VR, Mixed Reality, and future trends and innovations in AR/VR/MR. Human factor issues, user performance, sensorial conflict aspects of VR/AR/MR.

Text Books/References

1. Burdea, G. C. and P. Coffet. Virtual Reality Technology, Second Edition. Wiley-IEEE Press, 2003/2006.
2. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.
3. Alan Craig, William Sherman and Jeffrey Will, Developing Virtual Reality Applications, Foundations of Effective Design, Morgan Kaufmann, 2009.
4. Vladimir Geroimenko, Augmented Reality and Artificial Intelligence, Springer Nature Switzerland,2024
5. Gitanjali Rahul Shinde, Prashant Shantaram Dhotre, Parikshit Narendra Mahalle, Nilanjan Dey, Internet of Things Integrated Augmented Reality, Springer Nature Singapore,2020

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Project)	40	
End Semester Project Review		**30**

24CS737	BLOCKCHAIN TECHNOLOGY	2-0-2-3

Pre-Requisite(s): Basics of Computer Science
Course Type: Lab

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 the application and impact of blockchain technology in the financial industry and other industries

CO3: Evaluate security issues relating to blockchain and cryptocurrency

CO4: Design and analyze the impact of blockchain technology 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	2	3		

Syllabus

History, definition, features, types, and benefits of block chain and bitcoin, Consensus, CAP theorem and blockchain. Decentralization – methods, routes, smart contracts, platforms. Symmetric and Asymmetric cryptography - Public and private keys, theoretical foundations cryptography with practical examples.

Introduction to financial markets, use cases for block chain technology in the financial sector. Bitcoin, Transactions, Block chain, Bitcoin payments, technical concepts related to bitcoin cryptocurrency. Smart Contracts, definition of smart contracts, Ricardian contracts, Oracles, and the theoretical aspects of smart contracts.

Ethereum 101 - design and architecture of the Ethereum block chain, Various technical concepts related to the Ethereum block chain that explains the underlying principles, features, and

Components of this platform in depth. Hyperledger – protocol and architecture. Case studies on alternative Blockchains.

Text Books/References

1. Mastering Blockchain - Distributed ledgers, decentralization and smart contracts explained by Imran Bashir, Packt Publishing Ltd, Second Edition, 2018
2. Mastering Bitcoin: Unlocking Digital Cryptocurrencies by Andreas Antonopoulos, O'Reilly Publishing 2014.
3. Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher, Apress, First Edition, 2017.
4. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, Princeton University Press, 2016.
5. The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology by Vitalik Buterin, William Mougayar, Wiley; 1st edition, 2016.
6. Bitcoin: A Peer-to Peer Electronic Cash System by Satoshi Nakamoto, Online 2009, https://bitcoin.org/bitcoin.pdf .
7. Ethereum White Paper by Vitalik Buterin, Online 2017, https://ethereum.org/en/whitepaper/.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS738	CLOUD COMPUTING AND SERVICES	2-0-2-3

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

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 basic principles of Cloud Computing and Cloud Computing Architecture.

CO2: Analyze different types of cloud services – Delivery models, Deployment models.

CO3: Identify the significance of implementing virtualization techniques.

CO4: Apply cloud native application development for containerization and container orchestration.

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

Introduction to Cloud Computing, Distributed Computing Taxonomy – Cluster, Grid, P2P, Utility, Cloud, Edge, Fog computing paradigms; Introduction to Cloud Computing – Cloud delivery models (XaaS), Cloud deployment models (Private, Public, Hybrid); Characteristics of Cloud, Major use cases of Cloud; disadvantages and best practices; Major public cloud players in the market.

Security Issues and Challenges; Cloud Native application development – Introduction to JavaScript Cloud native application development. Public Cloud – Using public cloud for infrastructure management (compute and storage services), Web application deployment using

public cloud services, and Deploying container images in public cloud, Overview of cognitive services, Case study on architecting cloud-based solutions for a chosen scenario.

Virtualization – Microservices, Basics, Cloud vs Virtualization, Types of virtualization, Hypervisor types; Containers – Introduction to Dockers and containers, containerization vs virtualization, Docker architecture, Use cases, learning how to build container images, Operations on container images; Kubernetes – Need for orchestration, container orchestration methods, Introduction to Kubernetes, Kubernetes architecture, using YAML file, Running Kubernetes via minikube.

Text Books/References

1. Rajkumar Buyya et.al. Mastering cloud computing, McGraw Hill Education; 2013.
2. Cloud Computing: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti, Self published, Arshdeep Bahga & Vijay Madisetti (https://www.google.co.in/books/edition /Cloud_Computing_A_Hands_On_Approach/ETtmAgAAQBAJ?hl=en&gbpv=0)
3. Matthias K, Kane SP. Docker: Up & Running: Shipping Reliable Containers in Production. " O'Reilly Media, Inc."; 2018.
4. Kocher PS. Microservices and Containers. Addison-Wesley Professional; 2018.
5. Sarkar A, Shah A. Learning AWS: Design, build, and deploy responsive applications using AWS Cloud components. Packt Publishing Ltd; 2018.
6. Menga J. Docker on Amazon Web Services: Build, deploy, and manage your container applications at scale. Packt Publishing Ltd; 2018.
7. Bentley W. OpenStack Administration with Ansible 2. Packt Publishing Ltd; 2016.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS739	DEEP LEARNING	2-0-2-3

Pre-Requisite(s): None
Course Type: NLP

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 architectures.

CO2: Apply different frameworks to develop deep neural network architecture.

CO3: Apply different pre-trained and custom models to analyze the performance.

CO4: Design and deploy appropriate deep learning solutions for real-world problems.

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: Basics of image/signal like convolution - Perceptron - Multi Layer Perceptron - Feedforward Neural Networks - Back propagation, Training and Testing Neural Networks, Activation Functions, Loss Functions, Hyperparameters. Optimization algorithms and Regularization – Batch Normalization. Deep Neural Networks: Common Architectural Principles of Deep Networks, Building Blocks of Deep Networks.

Major Architectures of Deep Networks :Convolutional Neural Networks (CNNs), Deep CNN, Recurrent Neural Networks, Recursive Neural Networks, Adversarial learning, Transfer Learning, Generative AI.

Linear factor models, Autoencoders, Representation learning, Structured probabilistic models, Monte-Carlo models. Tensorflow/Pytorch: DenoisingSparsity in Autoencoders Models for Sequence Analysis –Vanishing Gradients Long Short-Term Memory (LSTM) Units- Primitives for RNN Models– Named Entity Recognition/Opinion Mining/Sentiment Analysis/Question Answering/Neural Summarization. Seminar : Cutting edge technologies like LLMs.

Text Books/References

1. Ian Goodfellow, YoshuaBengio and Aeron Courville, Deep Learning, MIT Press,First Edition, 2016.
2. Francois Chollet, Deep Learning with Python, Manning Publications Co, First Edition, 2018.
3. Gibson and Josh Patterson, Deep Learning A practitioner's approach, Adam O'Reilly, First Edition, 2017.
4. Nikhil Buduma, "Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithm", O'Reilly, 2017
5. Nikhil Ketkar, "Deep Learning with Python: A Hands-on Introduction", Apress, 2017.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS740	IMAGE AND VIDEO PROCESSING	2-0-2-3

Pre-Requisite(s): None
Course Type: Lab

Course Objectives

- To explore the algorithms in spatial and frequency domain relevant to image enhancement, restoration and segmentation applications.
- To understand the binary, gray scale and color image processing with real world applications
- To understand the video representation and motion analysis tools.

Course Outcomes

CO1: Understand the mathematical foundations of image and video processing.

CO2: Apply Spatial and frequency domain filtering for image enhancement.

CO3: Understand the fundamentals of Color Image Processing

CO4: Understand the principles in video representation and motion analysis.

CO-PO Mapping

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

Syllabus

Introduction-Digital Image Fundamentals: Elements of Computer Vision-Light and the Electromagnetic Spectrum-Light and the Electromagnetic Spectrum-Image Sensing and Acquisition-Image Sampling and Quantization-Some Basic Relationships between Pixels. Introduction to the Basic Mathematical Tools Used in Digital Image Processing- Intensity Transformations and Spatial Filtering: Basic Intensity Transformation Functions-Histogram Processing-Fundamentals of Spatial Filtering -Smoothing (Lowpass) Spatial Filters-Sharpening (High pass) Spatial Filters.

Filtering in the Frequency Domain: Sampling and the Fourier Transform of Sampled Functions-Discrete Fourier Transform of One Variable-Extensions to Functions of Two Variables-Properties of the 2-D and 3-D DFT and IDFT-Filtering in the Frequency

Domain-Image Smoothing Using Lowpass Frequency Domain Filters , Basics of morphological operators, Color Image Processing: Color Fundamental-Color Models-Color Transformations-Smoothing and Sharpening-Case Studies - Medical Imaging - Security and Surveillance Systems - Automated Driving Systems.

Video processing : Digital video, 2D and 3-D Feature Detection and Matching – points and patches, tomography, Motion estimation : Motion Models, Optical flow, Matching methods

Text Books/References

1. Rafael C. Gonzalez,Richard E. Woods,"Digital Image Processing", Pearson Education, 4th Edition, 2018
2. A. Murat Tekalp,"Digital Video Processing", O'Reilly, Second Edition, 2015
3. Szeliski R. Computer Vision: Algorithms and Applications Springer. New York. 2010. https://www.cs.ccu.edu.tw/~damon/tmp/SzeliskiBook_20100903_draft.pdf

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS741	INTERNET OF THINGS	2-0-2-3

Pre-Requisite(s): Advanced Networks
Course Type: Lab

Course Objectives

- Good hands-on exposure in 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 a specific use case.

Course Outcomes

CO1: Understand various concepts of IoT and related technologies.

CO2: Develop an IoT application using different hardware and software platforms

CO3: Implement various IoT Protocols in different layers

CO4: Design and deploy IoT applications for a given use case

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 IoT – IoT definition–Characteristics–IoT Complete Architectural Stack–IoT enabling Technologies–IoT Challenges. Sensors and Hardware for IoT–Hardware Platforms–Arduino, Raspberry Pi, MCU with wireless interfaces. A Case study with any one of the boards and data acquisition from sensors.

Protocols for IoT – Application level protocols: MQTT-CoAP, Web Sockets, Infrastructure protocol (IPV4/V6/RPL), Identification (URIs), Transport (Wi-Fi, Li-Fi, BLE), Discovery, Data Protocols, Device Management, Service Discovery and Management Protocols. IoT specific real-time database, UI design, IoT analytics, IoT privacy, security and vulnerability solutions. Integration of IoT end devices with Edge and Cloud Environment.

Case studies with architectural analysis: IoT applications–Smart City–Smart Water - Smart Agriculture–Smart Energy–Smart Healthcare–Smart Transportation–Smart Retail–Smart waste management – smart manufacturing.

Text Books/References

1. Pethuru Rajand Anupama C. Raman,"The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press, 2017.
2. Sudip Misra, Anandarup Mukherjee, Arijit Roy, “Introduction to IoT”, Cambridge University Press, 2020
3. Adrian McEwen,” Designing the Internet of Things”, Wiley,2013.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS742	OFFENSIVE CYBER SECURITY	2-0-2-3

Pre-Requisite(s): Basic knowledge of Networking protocols,Windows & Linux commands,Tools such as Bash Shell Scripting and Wireshark, SQL Commands, Oracle or MySQL Databases
Course Type: Lab

Course Objectives

- Analyze and evaluate suspicious entities leverage to steal data.
- Install, configure, utilize, and troubleshoot various offensive cyber security tools and software for proactive security strategies.

Course Outcomes

CO1: Understand the basic principles and features of Offensive Security and PenetrationTesting

CO2: Apply penetration testing strategies on various applications to evaluate and enhance these security of software systems

CO3: Identify and exploit common web and database vulnerabilities using tools

CO4: Comprehend Common Vulnerability Exposures, Common Weakness Enumeration of SANS Institute, and Critical Security Controls

CO5: Run Cyber Attacks, tests and probes in a practical context

CO-PO Mapping

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

Syllabus

Introduction to Offensive networking security, Penetration testing, and Ethical hacking–Types of PenetrationTesting: Network,Web Application,Wireless network, and Physical, Social Engineering.

Testing Methodologies: Blackbox / Whitebox– Intelligence Gathering: Passive Open-Source Information Gathering: OSINT Framework, Email Harvesting, Password Dumps, Maltego

Active Information Gathering: Port Scanning with Nmap, TCP/ UDP Scanning–Threat Modeling–Common Vulnerability Analysis: using Nessus, Banner Grabbing–Web Application Assessment: Exploitation, Burp Suite, SQL Injection, Cross-Site Scripting (XSS)–CVEs, National Vulnerability Database, CWE/SANSTOP25 Most Dangerous Software Errors, CIS:20 Critical Security Controls.

Buffer Overflows: Windows, Linux– Client- Side Attacks–Antivirus Circumvention– Privilege Escalation– Password Attacks– Active –Meta Sploit Framework–Red Team vsBlueTeam, Case Study: Mobile phone security anaylsis, rainbow crack project,, security configuration.

Text Books/References

1. Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali by OccupyTheWeb, Kindle Edition, No Starch Press, 2018.
2. Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, 1st Edition, Kindle Edition. No Starch Press, 2014.
3. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Edition, Wiley.com, 2011.
4. ames Forshaw, Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation, Kindle Edition, No Starch Press, 2017.
5. Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS743	PROJECT AND FINANCE MANAGEMENT	3-0-0-3

Pre-Requisite(s): Basic knowledge of Software Engineering
Course Type: Theory

Course Objectives

- Master the fundamentals and advanced techniques of project management, including planning, execution, risk management, and stakeholder communication, to successfully deliver projects on time and within budget.
- Develop a comprehensive understanding of financial management principles, including financial statement analysis, budgeting, and capital investment decisions, to effectively manage and allocate financial resources.
- Integrate project management and financial management skills to optimize organizational performance, enhance decision-making, and achieve strategic business objectives.

Course Outcomes

CO1: To acquire proficiency in planning, executing, and closing projects using industry-standard project management tools and methodologies.

CO2: To develop the ability to analyze financial statements, manage budgets, and make informed financial decisions using key financial management principles.

CO3: To enhance strategic decision-making skills by integrating project management and financial management practices to optimize organizational performance.

CO4: To evaluate methods, models, and technologies towards achieving project success.

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 Project Management- Software Projects - ways of categorizing software projects – problems with software projects - Project Life Cycle– Management -Setting objectives – Stakeholders – Project Charter - Project Scope- Project Organization - Project Team- Step-wise: An overview of project planning –project Evaluation. Fundamental Valuation Concepts: Time value of money – Risk and Return. Capital Budgeting: Techniques of capital

budgeting investment criteria– NPV – Benefit Cost Ratio – IRR – Payback Period – ARR – Investment appraisal in Practice – Estimation of Project cost flows.

Financial Analysis and Planning: financial instruments, sources of long-term, intermediate term and short-term finance. Analyzing Financial Performance – Break – even analysis and Leverages – Financial Planning and Budgeting. Software project Models - Tailoring of project models*- Selection of Appropriate Project Objectives- Software Effort Estimation Techniques, Function Point Analysis-Object Point -COCOMO. Activity planning-project schedules - sequencing and scheduling projects - Network planning model – AON and AOA-identifying critical activities-Crashing and Fast Tracking.

Risk management—Categories, Risk planning, Management and Control - Evaluating risks to the schedule - PERT- Resource Allocation – Project Management Plan- Monitoring and Tracking. Monitoring and control – Visualizing Progress, Earned Value Analysis, Closing a project - Managing people and organizing teams- organizational structures- Planning for small projects. Case Study: Agile Project Development*; 2. MS Project / Project Libre 3. PMBOK

Text Books/References

1. Mike Cotterell, Bob Hughes. Software Project Management, Fifth Edition, Tata McGraw-Hill; 2012.
2. Jalote P. Software Project Management in practice, Second edition, Person Education; 2003.
3. Chandra, P., Financial Management: Theory and Practice, 9e, TMH, 2017.
4. Eugene F. Brigham & Louis C. Gapenski, Financial Management – Theory and Practice,14e, 2015.

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS744	WIRELESS AND MOBILE NETWORKS	2–0-2-3

Pre-Requisite(s): Computer Networks
Course Type: Lab

Course Objectives

- Understand the basic operation and design of a wireless system.
- Understand and evaluate 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 and protocols of existing and emerging wireless networking technologies.

CO2: Understand the design principles of modern Wi-Fi standards, and analyze the applications using Wi-Fi.

CO3: Understand the design principles of modern bluetooth standards, analyze the applications using Bluetooth.

CO4: Understand the fundamentals of 4G and 5G cellular networks and design potential applications using 5G networks

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

Basic wireless propagation mechanisms-Reflection, diffraction and scattering, Free space Path loss model, basic terminologies (Receiver Sensitivity, Signal-to-Noise Ratio, Received Signal Strength), introduction to Multipath propagation (LoS, NLoS, Inter Symbol interference). Introduction to Modern wireless and mobile communication systems: Wireless Local Area Networks, Wireless Personal Area Networks, Cellular Networks- Principles of cellular networks, revolution of cellular networks 1G, 2G,3G,4G and 5G networks.

Modern Wireless Local Area Network: WiFi Basics, IEEE 802.11, ISM bands, IEEE 802.11 Channels, Physical Layer, Hidden node Problem, Collision Avoidance, IEEE 802.11 MAC Layer-CSMA/CA MAC Protocol, Architecture , Frame Format, Power Management, IEEE 802.11e(Enhanced QoS). Recent IEEE standards for WLAN-IEEE 802.11n, IEEE 802.11ac, IEEE 802.11ax.

Modern Wireless Personal Area Network : Introduction to Bluetooth and versions, Bluetooth1.1, Bluetooth 4.0-Bluetooth Low Energy (BLE), Bluetooth 5.0, Bluetooth 5.3. Modern Cellular Network: Multiple Access techniques-TDMA, FDMA and CDMA. 4G and 5G wireless system and standards, LTE-Network architecture and protocols, LTE Advanced, 5G. Case study : Wireless Sensing Hands on Experience: WiFi/Bluetooth Packets Capturing and analysis using Packet capture tool

Text Books/References

1. Hassan, M. (2022). Wireless and Mobile Networking (1st ed.). CRC Press. https://doi.org/10.1201/9781003042600
2. Agha, K. A., Pujolle, G., & Yahiya, T. A. (2016). Mobile and Wireless Networks (1st ed.). Wiley. Retrieved from https://www.perlego.com/book/997148/mobile-and - wireless-networks-pdf (Original work published 2016)
3. Wireless Communications: Principles and Practice, by Theodore S. Rappaport, Prentice Hall.
4. Stallings W, “Wireless Communications & Networks”. Pearson Education India; 2009.
5. 802.11n: A Survival Guide, by Matthew Gast, O'Reilly Media.
6. 802.11ac: A Survival Guide, by Matthew Gast, O'Reilly Media.
7. Pei Zheng, Larry L. Peterson, Bruce S. Davie, Adrian Farrel, “Wireless Networking Complete”, Morgan Kaufmann,2009.
8. Online Resources, Technical papers in course related topics and IEEE Standards documents
9. WLAN (IEEE 802.11) capture setup, https://wiki.wireshark.org/CaptureSetup/WLAN

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS745	SOFTWARE DEFINED SYSTEMS	2-0-2-3

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 algorithm design, software design architecture and principles used in software defined systems

CO3: Understand the orchestration and management strategies for software-defined systems, including the use of automation and DevOps practices.

CO4: Develop hands-on skills in setting up, configuring, and managing software-defined systems using various tools and frameworks.

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 and history, key concepts and terminologies, Software-defined infrastructures, Software-defined networking, Software-defined storage. Software defined networking: principles, architecture, components, OpenFlow and other protocols, case studies and applications.

Network Function Virtualization (NFV): architecture, components, virtual network functions (VNF), Usecases and case studies. Software Defined Storage: basics, architectures, models,

implementation strategies, case studies and applications. Software-Defined Data Centers (SDDC): concepts, components and technologies, implementation challenges, real-world applications.

Software defined systems (SDS): orchestration and management, management strategies, monitoring and performance, security considerations. Automation and DevOps: Role of automation in SDS, DevOps practices and tools, CI/CD pipelines, case studies. Hands-on labs on Mini SDN environment, implementing NFV scenarios, orchestrating a software defined datacenter etc. Tools / Frameworks: Red Hat OpenStack Platform (RHOSP)

Text Books/References

1. Oswald Coker, Siamak Azodolmolky, Software Defined Networking with OpenFlow, 2nd edition, 2017.
2. Paul Goransson, Chuck Black, Timothy Culver, Software Defined Networks: A Comprehensive Approach, 2016.
3. Ken Gray and Thomas D. Nadeau, Network Function Virtualization: Concepts and Applications, 2016.
4. Deze Zeng , Lin Gu , Shengli Pan , Song Guo "Software Defined Systems - Sensing, Communication and Computation", Springer Briefs in Computer Science – 2020.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS746	SECURE CODING	2-0-2-3

Pre-Requisite(s): Basic Knowledge of Programming Language (s), Database Management, Network, Server
Course Type: Lab

Course Objectives

- To learn secure programming practices, configuration of various tiers and layers involved in Software Development.
- Build secure software resilient to cyber attacks.

Course Outcomes

CO1: Understand the basics of secure programming.

CO2: Understand the most frequent programming errors leading to software vulnerabilities.

CO3: Identify and analyze security problems in software.

CO4: Understand and protect against security threats and software vulnerabilities.

CO4: Effectively apply their knowledge to the construction of secure software systems.

CO-PO Mapping

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

Syllabus

Secure Software Development: Principles of Software Security - Proactive Security development process, Secure Software Development Cycle (S-SDLC),A Risk Management Framework - A taxonomy of Coding Errors, Security Methodologies, Security Framework, Security Models

Defensive Coding Practices: Concepts and Techniques : Buffer Overrun, Format String Problems, Integer Overflow, and Injection flaws : SQL Injection, Command Injection, Failure to Handle Errors, Cross Site Scripting, Broken Authentication and Session Management, Magic URLs, Weak Passwords, Failing to Protect Data, Weak random numbers, improper use of

cryptography, Insecure Direct Object References, Insecure De-serialization, Security Misconfiguration, Information Leakage, Race Conditions, Poor Usability, Not Updating Easily, Executing with too much privilege, Failing to protect network traffic, improper use of PKI, trusting network name resolution.

Security code analysis and review: Code review with a tool (fortify, coverty etc), Code analysis Securing Server, Database, Network and their secure configuration, Firewalls. Case Study : Recent Software vulnerabilities due to insecure programming and how to prevent them during design and implementation. Tools : Azure Devops,, Gitlab CI/CD with security features, Jenkins with security plugins, Sonarcube, OWASP dependency Check, PMD.

Text Books/References

1. Paul, M. (2016). Official (ISC) 2 Guide to the CSSLP. CRC Press.
2. Seacord, R. (2013). Secure Coding in C and C++ (2nd Edition). SEI Series in Software Engineering
3. Howard, Michael, David LeBlanc, and John Viega. "24 Deadly Sins of Software Security." Programming Flaws and How to Fix Them (2010). McGraw-Hill Education
4. Ransome, J., & Misra, A. (2018). Core software security: Security at the source. CRC press.
5. Bishop, M. (2019). Computer Security(2 nd Edition). Addison-Wesley Professional.
6. McGraw, G. (2006). Software security: building security in (Vol. 1). Addison-Wesley Professional
7. John Veiga, Gary Mc Graw, “Building Secure Software: How to Avoid Security Problems the Right Way”, Addison-Wesley Professional Computing Series, 2001
8. Writing Secure Code, Michael Howard and David LeBlanc,Microsoft Press.
9. Buffer Overflow Attacks: Detect, Exploit, Prevent by Jason Deckar,Syngress.
10. Threat Modeling, Frank Swiderski and Window Snyder,Microsoft Professional.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS747	ROBOTICS AND CONTROL	2-0-2-3

Pre-Requisite(s): None
Course Type: Lab

Course Objectives

- To develop basic understanding of robot functions and configurations.
- To gain foundations of robot dynamics.
- To model and simulate robots for any given specifications.

Course Outcomes

CO1: Understand the architecture and components of robot systems. **CO2:** Develop kinematic models for various types of mobile robots. **CO3:** Apply maneuverability, workspace and motion control algorithms. **CO4:** Analyze programs and simulate robots in ROS.

CO-PO Mapping

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

Syllabus

Evolution of Robotics, Robot Controllers – Embedded Controllers, Hardware interfacing – Sensors – Position Sensors, Orientation Sensors, Vision Sensors, Signal Conditioning – Actuators – DC Motors, Stepper Motors, Servos, Pulse Width Modulation, driver circuits.

Robot Control – On/off Control, P, PI and PID Control, Velocity and Position Control – Multiple motor synchronization methods – Human Robot Interaction – Bluetooth, RF, IR, Wi-Fi controller – Robot to Robot communication.

Robotic Motion – Degrees of Freedom – forward kinematics – inverse kinematics – 3D Rotations – manipulator transformation – Sizing and torque calculations – localization – path planning – ROS Installation – basic ROS programming – simulation of 2-wheeled robots in ROS – Robot Arm simulation in ROS – simulation of flying Robots in ROS – Case studies: Mobile robots, industrial robots, UAVs.

Text Books/References

1. Thomas Braunl, “Embedded Robotics: From Mobile Robots to Autonomous Vehicles with Raspberry Pi and Arduino”, 4th Edition, Springer, 2022.
2. A.Koubaa, “Robot Operating System – The Complete Reference”, 1st Volume, Springer, 2016
3. John J. Craig, “Introduction to Robotics: Mechanics and Control”, 4th Edition, Pearson, 2018.
4. M.Ben-Ari, F.Mondada, “Elements of Robotics”, Springer, 2018.
5. W. Jacak, “Intelligent Robotic Systems: Design, Planning and Control”, Kluwer Academic Publishers, 2002.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS748	EVOLUTIONARY ROBOTICS	2-0-2-3

Pre-Requisite(s): Python/C++, Basics of Computer Science
Course Type: Lab

Course Objectives

- This course aims to provide theories, methods, and technologies for designing robots and artificial systems inspired by evolution, development, and learning.
- The course also aims to show how robotic models can help to understand biological systems.

Course Outcomes

CO1: Understand the essential concepts of simulated evolution, development and learning in the context of robotic design.
CO2: Apply Evolutionary Computation and learning for design and development of artificial adaptive systems.
CO3: Apply new tools for evolving robot body and brains
CO4: Evaluate the evolution and learning techniques for design of robots

CO-PO Mapping

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

Syllabus

Why Robots and Evolutionary Robotics – Short History of AI - Embodied Cognition. The tools of the Trade: Artificial Neural Network – Evolutionary Algorithms History: The First Years of Evolutionary Robotics

Minimal Cognition: Continuous Time Recurrent Neural Networks – Minimal Cognition – Active Categorical Perception. Locomotion: Legged Locomotion – Bipedal Locomotion. Challenges: Modularity - Genotype-to-Phenotype Map – NEAT/HyperNEAT Crossing the Reality Gap: Radical Envelope-of-Noise Hypothesis - GOLEM Project - Resilient Machines – Transferability. Scalability/Crowdsourcing: DotBot Project – Twitch Play Robotics

Collective Robotics: Swarm Robotics – Evolution of Communication. Evolving Bodies and Brains: First Attempts (Karl Sim's Works) - LSystem Robots – Why Evolve Bodies – Adaptive Robots – Soft Robots

Text Books/References

1. Stefano Nolfi (2021). Behavioral and Cognitive Robotics: An Adaptive Perspective. Open Access Book.
2. Patricia A. Vargas, Ezequiel A. Di Paolo, Inman Harvey and Phil Husbands (Eds.) (2014). The Horizons of Evolutionary Robotics. MIT Press.
3. Dario Floreano and Claudio Mattiussi (2008). Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. The MIT Press.
4. Rolf Pfeifer and Josh C. Bongard (2006). How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books). The MIT Press.
5. Additional readings from current research literature.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS749	KNOWLEDGE NETWORKS	3-0-0-3

Pre-Requisite(s): None
Course Type: Theory

Course Objectives

- To understand the structural properties and patterns of knowledge networks using metrics such as centrality measures, clustering coefficients, and degree distributions.
- To analyze the Integration of Machine Learning Techniques and Knowledge Networks.
- To understand the Knowledge Network Dynamics for collaboration Environments.

Course Outcomes

CO1: Understand the fundamental concepts of knowledge networks.

CO2: Gain practical skills in constructing and visualizing knowledge networks using appropriate tools and techniques.

CO3: Develop ability to perform statistical analyses of knowledge networks using recent graph learning algorithms to identify patterns and insights

CO4: Understand and apply knowledge network principles to related real world problems and applications.

CO5: Evaluate the ethical, policy, and social implications of deploying knowledge networks, ensuring responsible and fair use of knowledge networks.

CO-PO Mapping

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

Syllabus

Introduction - Types of knowledge networks, Applications of knowledge networks, basic network metrics- nodes, edges, degree distribution, centrality measures-degree centrality, Eigen vector centrality. Network density, clustering co-efficient. Knowledge Network Dynamics - Information Diffusion and Community Evolution.

Data collection and network construction - data preprocessing, network visualization techniques, Gephi and NetworkX. Community detection – Graph Partitioning, Girvan-Newman Algorithm, Louvain Method, Latent Dirichlet Allocation (LDA), Evaluation of Community Detection: Modularity Score, Normalized Mutual Information (NMI).

Machine learning and Knowledge networks - Graph Embeddings: Node2Vec, DeepWalk, GraphSAGE. Graph Neural Networks (GNNs): Graph Attention Networks (GATs), Graph Autoencoders. Link Prediction using Unsupervised Methods: Using similarity measures (e.g., Jaccard coefficient, Adamic/Adar index). Node Classification: Semi-Supervised Learning, Label Propagation. Knowledge Graph Completion- BERT-based Models. Event Detection- Change Point Detection, Burst Detection. Case study: Connected papers and Google Knowledge Graph tools.

Text Books/References

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

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS750	FOUNDATIONS OF DATA SCIENCE	2-0-2-3

Pre-Requisite(s): Basic Probability
Course Type: Lab

Course Objectives

- Statistical foundations of data science.
- Techniques to pre-process raw data; (data wrangling, munging) with Numpy, Pandas and other Python statistical packages; visualization with Matplotlb, Plotly and Bokeh; EDA; statistical inferences
- Predictions using statistical tests
- Estimation of statistical parameters
- Introduction to Time Series.

Course Outcomes

CO1: Understand the statistical foundations of data science.

CO2: Apply pre-processing techniques over raw data, conduct exploratory data analysis, create insightful visualizations, and identify patterns to enable further analysis

CO3: Identify machine learning algorithms for prediction/classification and to derive insights

CO4: Analyze the degree of certainty of predictions using statistical test and models.

CO5: Explore the statistical foundations of time series, and employ basic ARIMA models for time series prediction.

CO-PO Mapping

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

Syllabus

Introduction to Data Science, Causality and Experiments, Python libraries for data wrangling: Basics of Numpy arrays, aggregations, computations on arrays, fancy indexing, structured arrays. Data Pre-processing - Data manipulation with Pandas – Data indexing and selection – Operating on data – Missing data – Hierarchical indexing – Combining datasets – Aggregation and

grouping – Pivot tables - Data cleaning – Data reduction - Data transformation. Visualization and Graphing: Visualizing Categorical Distributions - Visualizing Numerical Distributions-Overlaid Graphs and plots- Summary statistics of exploratory data analysis- Randomness- Probability- Introduction to Statistics. Ethics and privacy.

Sampling, Sample Means and Sample Sizes. Probability distributions and density functions (univariate and multivariate), Error Probabilities; Expectations and moments; Covariance and correlation; Sampling and Empirical distributions; Permutation Testing, Statistical Inference; Central Limit Theorem, Hypothesis testing of means, proportions, variances and correlations - Assessing Models - Decisions and Uncertainty, Comparing Samples - A/B Testing, P-Values, Causality. Estimation - Resampling and Bootstrap - Confidence Intervals, Properties of Mean - Variability of mean -Choosing Sample Size - Graphical Models. Case Studies

Time Series Analysis: Time Series patterns - Statistical fundamentals – Descriptive statistics - Measuring errors – Correlation and Covariance – Autocorrelations –ACF and PACF- Stationarity- Durbin-Watson Statistic-Overview of Univariate methods - Moving averages; WMA; Exponential smoothing - ARIMA model identification- ARIMA(1,0,0)- ARIMA(0,0,1)– ARIMA(0,1,0)- Akaike Information Criterion – Schwarz Bayesian Information Criterion.

Text Books/References

1. AniAdhikari and John DeNero, ”Computational and Inferential Thinking: The Foundations of Data Science”, e-book.
2. Joel Grus, ”Data Science from Scratch: First Principles with Python”, 2/e, O’Reilly Media, 2019.
3. Jake VanderPlas, Python Data Science Handbook: Essential Tools For Working With Data, O’Reilly Media, 2016.
4. Wes McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter 3rd Edition, O’Reilly Media, 2022.
5. Cathy O’Neil and Rachel Schutt,”Doing Data Science”, O’Reilly Media, 2013.
6. Stephen A. Delurgio. Forecasting Principles and Applications, McGraw-Hill International Editions; 1998.
7. Jason Brownlee, Introduction to Time Series forecasting with Python, 2017.

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS751	BIG DATA ANALYTICS	3-0-0-3

Pre-Requisite(s): None
Course Type: Theory

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 fundamental concepts of Big Data and its significance in modern data-driven environments.

CO2: Apply various Big Data technologies and tools to effectively store, retrieve, and process Big Data.

CO3: Analyze and visualize large datasets to extract meaningful insights and support decision making processes.

CO4: Design and develop machine learning algorithms for predictive analytics and Big Data based frameworks.

CO5: Utilize functional programming and distributed computing principles to optimize 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

Characteristics of Big Data, Types of Big Data, Technologies for Big Data, Infrastructure for Big Data, Use of Data Analytics, Big Data Challenges, NoSQL, Comparison of SQL and NoSQL, Distributed Computing Challenges, Hadoop Ecosystem: HDFS (Hadoop Distributed File System), MapReduce: Inputs, Outputs, and Data Serialization, Managing Resources with Hadoop YARN, Interacting with the Hadoop Ecosystem, Functional Programming in Scala:

Basic Syntax, Type Inference, Parameters, Recursive Arbitrary Collections, ConsList, Arrays, Tail Recursion, Higher-Order Functions.

MapReduce Programming: Mapper, Reducer, Combiner, Partitioner, Real-Time MapReduce Applications, Data Serialization, Apache Spark: Resilient Distributed Datasets (RDDs), Creating RDDs, Lineage and Fault Tolerance, DAGs, Immutability, Task Division and Partitions, Transformations and Actions, Lazy Evaluations and Optimization, Formatting and Housing Data from Spark RDDs, Persistence

Hive Architecture: Hive Data Types, Hive File Format, Hive Query Language (HQL), User-Defined Functions (UDF) in Hive, Introduction to Machine Learning with Spark: MLlib, Building a Machine Learning Pipeline in Spark, Pig on Hadoop: Anatomy of Pig, Use Cases for Pig, ETL Processing, Data Types in Pig, Running Pig, Execution Modes of Pig, HDFS Commands, Relational Operators, Piggy Bank.

Text Books/References

1. Seema Acharya, Subhashini Chellappan, "Big Data and Analytics", Wiley Publication, 2015.
2. Hurwitz JS, Nugent A, Halper F, Kaufman M. "Big Data for Dummies", John Wiley & Sons, 2013.
3. Tom White, "Hadoop: The Definitive Guide", O'Reilly Publications, 2011.
4. Kyle Banker, "Mongo DB in Action", Manning Publications Company, 2012.

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS752	HIGH PERFORMANCE COMPUTING	2-0-2-3

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 HPC Systems and Technologies for real world applications
- To evaluate and innovate in HPC Design trough Data Centre Environments

Course Outcomes

CO1: Understand and apply various levels of parallelism including instruction, transaction, task, thread, memory, function, and data flow models.
CO2: Implement and Optimize Parallel Programs with use of Modern Tools.
CO3: Analyze and compare scalable storage systems including RAID, SSD cache, SAS, and SAN.
CO4: Evaluate the potential and limitations of emerging HPC technologies and their applications in various domains

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 HPC architectures: Shared and Distributed memory architectures, Multiprocessor Architecture. Parallel Processing Concepts; Levels and model of parallelism: instruction, transaction, task, thread, memory, function, data flow models, demand-driven computation; Parallel architectures: superscalar architectures, multi-core, multi-threaded, server and cloud; Fundamental design issues in HPC: Load balancing, scheduling, synchronization and resource management.

Operating systems for scalable HPC; Parallel languages and programming environments; OpenMP, Pthread, MPI, java, Cilk; Performance analysis of parallel algorithms; Fundamental

limitations in HPC: bandwidth, latency, and latency hiding techniques; Benchmarking HPC: scientific, engineering, commercial applications and workloads.

Scalable storage systems: RAID, SSD cache, SAS, SAN; HPC based on cluster, cloud, and grid computing: economic model, infrastructure, platform, computation as service; Accelerated HPC: architecture, programming and typical accelerated system with GPU, FPGA, Xeon Phi, Cell BE; Power-aware HPC Design: computing and communication, processing, memory design, interconnect design, power management; Advanced topics: peta scale computing; big data processing, optics in HPC, quantum computers. Case Studies on the parallel machine and HPC cluster using Pthread, OpenMP, MPI, Nvidia Cuda and Cilk.

Text Books/References

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

Evaluation Pattern: 70/30

Assessment	Internal Weightage	External Weightage
Midterm Examination	20	
Continuous Assessment (Theory)	10	
Continuous Assessment (Lab)	40	
End Semester		**30**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

24CS753	MEDICAL IMAGING	3-0-0-3

Pre-Requisite(s): None
Course Type: Theory

Course Objectives

- The course introduces various modalities of medical imaging.
- Covers image processing algorithms specific for medical imaging such as registration and fusion.

Course Outcomes

CO1: Understand various medical imaging modalities.

CO2: Apply image registration techniques to medical images.

CO3: Apply image fusion algorithms to medical images.

CO4: Apply medical image analysis algorithms for Computer Aided Diagnosis.

CO-PO Mapping

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

Syllabus

Basics of Medical Image Source, Medical Image Representation, Image Quality and Information Content, Operations In Intensity Space, Image Registration. Image Registration Methods Principal Axes Registration, Multiresolution Registration, Optimization-Based Registration, Boundary Registration , Model-Based Registration , Adaptive Registration

Image Fusion, Analysis of Texture, Analysis of Oriented Patterns, Image Reconstruction from Projections. Medical Image Fusion Local Weighted Voting ,MV Algorithm ,Global Weighted Voting , Semi-Local Weighted, Patch-Based Local Weighted Voting Segmentation Algorithm , Patch-Based Global Weighted Fusion Segmentation Algorithm

Removal of Artifacts, Image Enhancement, Detection of Regions of Interest, Deconvolution, Deblurring, And Restoration; Analysis Of Shape- Case studies in Computer Aided Diagnosis, CNN for medical image analysis. Case studies based on the latest advancements in Medical Image Processing : Detection, Segmentation, and Decision Making

Text Books/References

1. Birkfellner W. Applied medical image processing: a basic course. CRC Press; 2014.
2. Rangaraj M. Rangayyan.Biomedical Image Analysis,First Edition,CRC Press;2004.
3. Jan J. Medical image processing, reconstruction and restoration: concepts and methods. Crc Press; 2005.
4. S. Kevin Zhou, Hayit Greenspan, Dinggang Shen, Deep Learning for Medical Image Analysis, 2024

Evaluation Pattern: 60/40

Assessment	Internal Weightage	External Weightage
Midterm Examination	30	
Continuous Assessment	30	
End Semester		**40**

Note: Continuous assessments can include quizzes, tutorials, lab assessments, case study and project reviews. Midterm and End semester exams can be a theory exam or lab integrated exam for two hours

23HU601	Career Competency I	L-T-P-C: 0-0-3-P/F

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

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

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 gab to improve their debating skills.

Writing Skills: Introduce formal written communication and keep the students informed about the etiquettes of email writing. Make students practise 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:

- Andrew J DuRbin , "Applied Psychology: Individual and organizational effectiveness", Pearson- Merril Prentice Hall, 2004
- Michael G Aamodt, "An Applied Approach, 6th edition", Wadsworth Cengage Learning, 2010

Assertiveness skills:

- Robert Bolton, Dorothy Grover Bolton, "People Style at Work..and Beyond: Making Bad Relationships Good and Good", Ridge Associates Inc., 2009
- John Hayes "Interpersonal skills at work", Routledge, 2003
- 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

Self-perception and self-confidence:

- Mark J Martinko, "Attribution theory: an organizational perspective", St. Lucie, 1995
- Miles Hewstone, "Attribution Theory: Social and Functional Extensions", Blackwell, 1983

Time management:

- Stephen Covey, "The habits of highly effective people", Free press Revised edition, 2004
- 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
- Kenneth H. Blanchard and Spencer Johnson, "The One Minute Manager" , William Morrow, 1984

Verbal:

- Erica Meltzer, "The Ultimate Guide to SAT Grammar"
- Green, Sharon, and Ira K. Wolf, "Barron's New GRE", Barron's Educational Series, 2011
- Jeff Kolby, Scott Thornburg & Kathleen Pierce, "Nova's GRE Prep Course"
- Kaplan, "Kaplan New GRE Premier", 2011-2012
- Kaplan's GRE Comprehensive Programme
- Lewis Norman, "Word Power Made Easy", Goyal Publishers, Reprint edition, 1 June 2011
- Manhattan Prep, "GRE Verbal Strategies Effective Strategies Practice from 99th Percentile Instructors"

- Pearson- “A Complete Manual for CAT”, 2013
- R.S. Aggarwal, “A Modern Approach to Verbal Reasoning”
- S. Upendran, “Know Your English”, Universities Press (India) Limited, 2015
- Sharon Weiner Green, Ira K. Wolf, “Barron's New GRE, 19th edition (Barron's GRE)”, 2019
- Wren & Martin, “English Grammar & Composition”
- www.bbc.co.uk/learningenglish
- www.cambridgeenglish.org
- www.englishforeveryone.org
- www.merriam-webster.com

Aptitude:

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

23HU611	Career Competency II	L-T-P-C: 0-0-3-1

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.

C05: 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

C06: 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, Cryptarithmetic 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

- Thomas L.Quick, "Successful team building", AMACOM Div American Mgmt Assn, 1992
- **Brian Cole Miller, "Quick Team-Building Activities for Busy Managers: 50 Exercises That Get Results in Just 15 Minutes", AMACOM; 1 edition, 2003.**
- **Patrick Lencioni, "The Five Dysfunctions of a Team: A Leadership Fable", Jossey-Bass, 1st Edition, 2002**

Verbal

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

Aptitude

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

22AVP103 Mastery Over Mind (MAOM) 1-0-2 2

1. Course Overview

Master Over the Mind (MAOM) is an Amrita initiative to implement schemes and organize 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.

3. Evaluation and Grading

Internal			External	Total
Components	Weightage		Practical (attendance and class participation) 60%	100%
Quizzes(based on the reading material)	20%	40%		
Assignments (Based on webinars and lecture series)	20%			

4. Course Outcomes (CO)

CO1: Relate to the causes of stress in one's life.
CO2: Experiment with a range of relaxation techniques
CO3: Model a meditative approach to work, study, and life.
CO4: Develop appropriate practice of MA-OM technique that is effective in one's life
CO5: Inculcate a higher level of awareness and focus.
CO6: Evaluate the impact of a meditation technique

***Program Outcomes (PO) (As given by NBA and ABET)**
PO1: Engineering Knowledge
PO2: Problem Analysis
PO3: Design/Development of Solutions
PO4: Conduct Investigations of complex problems
PO5: Modern tools usage
PO6: Engineer and Society
PO7: Environment and Sustainability
PO8: Ethics
PO9: Individual & Team work
PO10: Communication
PO11: Project management & Finance
PO12: Lifelong learning

CO – PO Affinity Map

P O/ C O	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2	PSO3
CO 1	3	3	3	2		-	2	3	-	3	-	3	-	-	-
CO 2	3	3	3	2	2	_	2	3	3	3	-	3	-	-	-
CO 3	3	3	2	2	2	2	2	3	3	3	-	3	-	-	-
CO 4	3	3	3	2	-	2	3	3	3	3	-	3	-	-	-
CO 5	3	2	2	2	-	2	-	3	2	2	-	2	-	-	-
CO 6	3	2	2	2	3	2	_	3	2	2	-	2	-	-	-