DEPARTMENT OF COMPUTER APPLICATIONS
NATIONAL INSTITUTE OF TECHNOLOGY
TIRUCHIRAPPALLI - 620015

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

CURRICULUM

(EFFECTIVE FROM 2024-2025)

VISION OF THE INSTITUTE

To be a university globally trusted for technical excellence where learning and research integrate to sustain society and industry.

MISSION OF THE INSTITUTE

- To offer undergraduate, postgraduate, doctoral and modular programmes in multi-disciplinary / inter-disciplinary and emerging areas.
- To create a converging learning environment to serve a dynamically evolving society.
- To promote innovation for sustainable solutions by forging global collaborations with academia and industry in cutting-edge research.
- To be an intellectual ecosystem where human capabilities can develop holistically.

VISION OF THE DEPARTMENT

- Towards a school of Information Science and Technology conforming to international standards

MISSION OF THE DEPARTMENT

- To offer state-of-art education in Information Science and Technology
- To provide strong theoretical foundation complemented with extensive practical training
- To inculcate value-based, socially committed professionalism to the cause of overall development of students and society

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1	Prepare graduates to become computer professionals with comprehensive knowledge and skills to produce software for emerging requirement
PEO2	Prepare graduates to become continuous learner with aptitude for teaching and research with societal focus
PEO3	Prepare graduates to become Consultant / Entrepreneurs in the IT and ITES industries with confidence in self-employment

PROGRAMME OUTCOMES (POs)

PO1	An ability to independently carry out research /investigation and development work to solve practical problems
PO2	An ability to write and present a substantial technical report/document
PO3	Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program, higher than the requirements in the appropriate bachelor program

CURRICULUM FRAMEWORK / FLEXIBLE CURRICULUM / NEP 2020 / M.Sc.

Components	Number of Courses	Number of Credits
Programme Core (PC)	6 / Year	42
Programme Elective (PE)	6 / Year	
Essential Laboratory Requirements (ELR)	3 / Year	6
Internship / Industrial Training / Academic Attachment (I/A)	1	2
Open Elective (OE)	2	6
Project Phase-I	1	12
Project Phase-II	1	12
Total	**20**	**80**

CURRICULUM

SEMESTER I

Code	Course of Study	Credit
CAS711	Mathematical Foundations of Computer Sciences	4
CAS713	Computer Organization and Architecture	4
CAS715	Data Structures and Algorithms	3
CAS717	Database Management Systems	3
CAS719	Operating Systems Fundamentals	4
CAS7AX	Elective I	4
CAS701	Data Structures and Algorithms Lab using C/C++	2
CAS703	Database Management Systems Lab	2
	TOTAL	**26**

SEMESTER II

Code	Course of Study	Credit
CAS712	Networking Technologies	3
CAS7BX	Elective II	4
CAS7CX	Elective III	4
CAS7DX	Elective IV	3
CAS7EX	Elective V (OE)	3
CAS7FX	Elective VI	3
CAS702	Internet Programming Lab	2
	TOTAL	**22**

SUMMER TERM (evaluation in the III semester)

Code	Course of Study	Credit

CAS748	Internship / Industrial Training / Academic Attachment (I/A) (6 weeks to 8 weeks)	2

SEMESTER III

Code	Course of Study	Credit
CAS749	Project Phase-I	12

SEMESTER IV

Code	Course of Study	Credit
CAS750	Project Phase-II	12

ONLINE COURSE (OC) (To be completed between I to IV semester)

SI No.	Code	Course of Study	Credit
1.		Online course 1	3
2.		Online course 2	3
			6

PROGRAMME ELECTIVES (PE)

ELECTIVE - I

Sl. No.	Code	Course of Study	Credit
1.	CAS7A1	Artificial Intelligence	4
2.	CAS7A2	Cryptography	4
3.	CAS7A3	Distributed Computing	4
4.	CAS7A4	Soft Computing Techniques	4

ELECTIVE - II

Sl. No.	Code	Course of Study	Credit
1.	CAS7B1	Digital Image Processing	4

2.	CAS7B2	Theory of Computation	4
3.	CAS7B3	Fault Tolerance Systems and Techniques	4
4.	CAS7B4	Mobile Computing	4

ELECTIVE - III

Sl. No.	Code	Course of Study	Credit
1.	CAS7C1	GPGPU Programming	4
2.	CAS7C2	Software Engineering	4
3.	CAS7C3	Computer Vision and Pattern Recognition	4
4.	CAS7C4	Web Applications Development	4

ELECTIVE - IV

Sl. No.	Code	Course of Study	Credit
1.	CAS7D1	Defensive and Secure Software Development	3
2.	CAS7D2	Software Verification and Validation	3
3.	CAS7D3	Machine Learning and Deep Learning Techniques	3
4.	CAS7D4	Compiler Design	3

ELECTIVE - V

Sl. No.	Code	Course of Study	Credit
1.	CAS7E1	Big Data Analytics	3
2.	CAS7E2	Quantum Computing	3
3.	CAS7E3	Mobile Applications Development	3
4.	CAS7E4	Augmented Reality and Virtual Reality	3

ELECTIVE - VI

Sl. No.	Code	Course of Study	Credit
1.	CAS7F1	Blockchain Technologies	3
2.	CAS7F2	Design Patterns	3
3.	CAS7F3	Internet of Things	3
4.	CAS7F4	Real Time Systems	3

SEMESTER I-IV

Code	Course of Study	Credit
	Open Elective I	3
	Open Elective II	3
		6

OPEN ELECTIVE (OE) OFFERED BY THE DEPARTMENT

Sl. No.	Code	Course of Study	Credits
1.	CAS7E1	Big Data Analytics	3

COURSE OUTCOME AND PROGRAMME OUTCOME MAPPING

PROGRAMME CORE (PC)

Course Outcomes: On successful completion of the course, students will be able to:

Course Code	Course Title	CO	Course outcomes At the end of the course student will be able	PO1	PO2	PO3
CAS711	Mathematical Foundations of Computer Sciences	CO1	Understand sets, functions and relations and apply these discrete structures in different areas of computing.	1	1	2
		CO2	Analyse and apply logical propositions to test the validity of arguments.	2	2	3
		CO3	Apply the concepts of trees and graphs to solve real world problems	2	2	3
		CO4	Apply the concepts of probability and statistics to solve machine learning and data science problems	2	3	3
CAS713	Computer Organization and Architecture	CO1	Understand basic structure and operation of digital computer.	3	-	3
		CO2	Understand the design of control unit.	3	-	3
		CO3	Understand the design of ALU to perform arithmetic and logic operations on fixed point and floating numbers.	3	-	3
		CO4	Understand different types of instructions and addressing modes supported in the instruction set of CPUs.	3	-	3
CAS715	Data Structures and Algorithms	CO1	Develop own ADTs according to the nature of problems.	3	-	3
		CO2	Design algorithms using various	3	2	3

			strategies.			
		CO3	Compute time- and space complexities of various algorithms.	3	-	3
CAS717	Database Management Systems	CO1	Construct SQL Queries using relational algebra	2	-	3
		CO2	Design database using ER model and normalize the database	3	-	3
		CO3	Construct queries to handle transaction processing and maintain consistency of the database	3	2	3
		CO4	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database	3	2	3
		CO5	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.	3	2	3
CAS719	Operating Systems Fundamentals	CO1	Use system calls for managing processes, memory, and the file system.	3	-	3
		CO2	Explore the functionalities of distributed and mobile operating systems.	2	1	3
		CO3	Explore various types of operating systems including UNIX, Linux, Android, and Windows.	3	1	3
CAS712	Networking Technologies	CO1	Describe available LAN and WAN Technologies.	3	-	3
		CO2	Describe the principles of packet switching, forwarding, and routing.	3	-	3
		CO3	Distinguish between TCP and UDP packet formats.	3	-	3

		CO4	Describe the available application protocols and networking services.	3	-	3

LABORATORY

Course Code	Course Title	CO	Course Outcomes At the end of the course student will be able	PO1	PO2	PO3
CAS701	Data Structures and Algorithms Lab using C/C++	CO1	Write C/C++ programs for problem solving.	3	3	3
		CO2	Implement linear and nonlinear data structures to solve real-time problems.	3	3	3
		CO3	Perform searching and sorting techniques on different application domains.	3	3	3
		CO4	Implement different algorithm design strategies to solve complex problems.	3	1	3
CAS703	Database Management Systems Lab	CO1	Construct SQL Queries using relational algebra.	3	3	3
		CO2	Design database using ER model and normalize the database.	3	3	3
		CO3	Construct queries to handle transaction processing and maintain consistency of the Database.	3	3	3
		CO4	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the	3	1	3

			database.			
CAS702	Internet Programming Lab	CO1	Use and explore simulation tools for networking configuration, application, applying algorithms.	3	2	3
		CO2	Develop web pages with mark-ups, style sheets, multimedia elements.	3	2	3
		CO3	Design and implement real time applications using JavaScript.	3	2	3
		CO4	Design and implement client server applications using databases, server-side technologies	3	2	3

PROGRAMME ELECTIVES (PE)

Course Code	Course Title	CO	Course outcomes	PO1	PO2	PO3
CAS7A1	Artificial Intelligence	CO1	Compare between AI and non-AI solutions.	3	-	3
		CO2	Apply AI techniques in problem-solving.	3	2	3
		CO3	Analyse the best search technique and implement it in real-life applications.	3	1	3
		CO4	Classify supervised and unsupervised learning and knowledge representation.	3	1	3
CAS7A2	Cryptography	CO1	Explain basic components of information systems security.	2	1	1
		CO2	Design and use symmetric ciphers to protect data at rest and in transit.	2	3	3
		CO3	Design and use public key cryptosystem to protect data at rest and in transit.	2	3	3
		CO4	Design and use Blockchain-based security system to protect	2	3	3

			information systems from ever-growing Cyber-attacks			
CAS7A3	Distributed Computing	CO1	Explain the concepts and technologies of distributed systems.	2	1	3
		CO2	Learn the concepts of designing and developing web services.	3	1	3
		CO3	Understand cloud computing architecture and techniques.	3	-	3
		CO4	Analyze and evaluate cloud computing models.	3	-	3
CAS7A4	Soft Computing Techniques	CO1	Explain the concepts and technologies of distributed systems.	3	-	3
		CO2	Learn the concepts of designing and developing web services.	3	-	3
		CO3	Understand cloud computing architecture and techniques.	3	-	3
CAS7B1	Digital Image Processing	CO1	Know and understand the basics and fundamentals of digital image processing, such as digitization, sampling, quantization, and 2D-transforms.	3	1	3
		CO2	Operate on images using the techniques of smoothing, sharpening and enhancement.	3	1	3
		CO3	Understand the restoration concepts and filtering techniques.	3	-	3
		CO4	Learn the basics of segmentation, features extraction, compression and recognition methods for color models.	3	-	3
CAS7B2	Theory of Computation	CO1	Define languages by abstract, recursive definitions and regular expressions.	2	-	3
		CO2	Design a finite automaton to recognize a given regular language.	3	-	3
		CO3	Transform a language into regular expression or finite automaton or transition graph.	3	-	3
		CO4	Define deterministic finite automata and nondeterministic finite automata.	2	-	2

CAS7B3	Fault Tolerance Systems and Techniques	CO1	Understand the basic concepts of fault tolerance	3	-	2
		CO2	Practice the application principles of the fault tolerance	3	-	3
		CO3	Analyse the fault tolerance in hardware and software level	3	-	3
CAS7B4	Mobile Computing	CO1	Understand concepts of Mobile Communication.	3	-	3
		CO2	Analyze the next generation Mobile Communication System.	2	1	3
		CO3	Understand network and transport layers of Mobile Communication.	2	-	3
		CO4	Analyze various protocols of all layers for mobile and ad hoc wireless communication networks.	3	-	3
CAS7C1	GPGPU Programming	CO1	Analyze GPU Design for finding solutions.	3	-	3
		CO2	Work with parallel programming paradigms.	3	2	3
		CO3	Solve high performance computing problems using GPUs.	3	2	3
CAS7C2	Software Engineering	CO1	Demonstrate a basic understanding of software engineering practices from vision to analysis, design, development, validation, deployment and maintenance.	3	2	3
		CO2	Develop skills to create and use various software Engineering based techniques and tools to solve real world problems.	3	3	3
		CO3	Estimate cost, effort and risk involved in a software project development	3	3	3
CAS7C3	Computer Vision and Pattern Recognition	CO1	Apply fundamental algorithms in Image Processing and analyze their applicability for real time problems.	3	-	3
		CO2	Design solutions for various computer vision and pattern recognition problems	3	-	3
		CO3	Design and develop innovative computer vision applications or systems.	3	2	3
		C01	Understand significance and	3	1	3

CAS7C4	Web Applications Development		features of Web Applications.			
		C02	Apply Markups, Style Sheets, XML technologies in designing websites and developing web applications.	3	2	3
		C03	Design and develop client side applications using javascript for real time scenarios.	3	2	3
		C04	Build and deploy client-server applications using databases and server side technologies.	3	3	3
CAS7D1	Defensive and Secure Software Development	CO1	Understand software security fundamentals.	3	-	3
		CO2	Develop secure software.	3	2	3
		CO3	Use secure coding practices in software development.	3	-	3
CAS7D2	Software Verification and Validation	CO1	Understand the basic concepts of software engineering.	3	2	3
		CO2	Practice the application principles of Software testing.	3	3	3
		CO3	Analyze the process of software verification and validation	3	-	3
CAS7D3	Machine Learning and Deep Learning Techniques	CO1	Select real-world applications that need machine learning based solutions.	1	2	2
		CO2	Implement and apply machine learning algorithms.	2	2	3
		CO3	Select appropriate algorithms for solving a particular group of real-world problems.	1	2	2
		CO4	Recognize the characteristics of machine learning techniques that are useful to solve real-world problems.	2	2	3
CAS7D4	Compiler Design	CO1	Explain various phases of compiler.	3	-	3
		CO2	Explain various parsing techniques.	3	-	3
		CO3	Explain the process of intermediate code generation.	3	-	3
		CO4	Explain various code optimization techniques.	3	-	3
CAS7E1	Big Data Analytics	CO1	Comprehend the concepts of big data analytics.	3	-	3
		CO2	Build web-intelligence	3	2	3

			applications exploiting big data using big data platforms based on the 'map-reduce' parallel programming framework.			
		CO3	Effectively use NoSQL databases for storage and retrieval of big data.	3	2	3
CAS7E2	Quantum Computing	CO1	Basics of complex vector spaces	1	1	2
		CO2	Quantum mechanics as applied in Quantum computing	1	2	3
		CO3	Architecture and algorithms	2	2	3
		CO4	Fundamentals of Quantum computations.	1	2	3
CAS7E3	Mobile Applications Development	CO1	Describe the requirements for mobile applications.	3	2	3
		CO2	Develop design for mobile applications for specific requirements.	3	2	3
		CO3	Implement the design using mobile application development frameworks.	3	2	3
CAS7E4	Augmented Reality and Virtual Reality	CO1	Understand the principles and use cases of AR and VR.	3	1	3
		CO2	Understand various I/O interfaces – functionalities and usage.	3	1	3
		CO3	Explore the visual computations present in AR and VR applications.	3	-	3
		CO4	Gain knowledge over various tools and frameworks for AR and VR applications development	3	2	3
CAS7F1	Blockchain Technologies	CO1	Explain design principles of Bitcoin and Ethereum.	3	-	3
		CO2	Explain the Simplified Payment Verification protocol.	3	1	3
		CO3	Interact with a blockchain system by sending and reading transactions.	2	-	3
		CO4	Design, build, and deploy a distributed application.	3	2	3
CAS7E2	Design Patterns	CO1	Solve common problems in software design with ease.	3	-	3
		CO2	Analyze object-oriented design for patterns.	3	2	3
		CO3	Represent design decisions more	3	2	3

			effectively with examples and architectural use cases.			
CAS7E3	Internet of Things	CO1	To understand the working of IoT devices.	3	-	3
		CO2	To Program the IoT devices.	3	2	3
		CO3	To develop applications using IoT devices.	3	2	3
CAS7E4	Real Time Systems	CO1	Gain knowledge about Schedulability analysis.	3	-	3
		CO2	Learn about the Real-time programming environments.	3	2	3
		CO3	Attain knowledge about real time communication and databases.	3	-	3
		CO4	Develop a real time system.	3	2	3

3 - High; 2 - Medium; 1 - Low

SYLLABUS

SEMESTER I

Course Code	**:**	CAS711
Course Title	**:**	Mathematical Foundations of Computer Sciences
Type of Course	**:**	PC
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods	**:**	Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To acquire skills in solving mathematical and logical problems
CLO2	To comprehend mathematical principles and logic.
CLO3	To understand fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science.

Course Content

Set Theory: Sets and operations, properties - power set - methods of proof - relations, graph, and matrix of a relation - partial and total orders, well ordering - equivalence relations, classes and properties - functions, 1-1, onto and bijective - composition of relations and functions - inverse functions.

Mathematical Logic: Propositions and logical operators – Truth table – Equivalences and implications – Basic laws– Some more connectives – Functionally complete set of connectives – Review of Propositional Calculus - Validity - Satisfiability related concepts - CNF and DNF forms - Conversion of arbitrary propositional formula to CNF or DNF.

Graph Theory: Definitions and basic results - Representation of a graph by a matrix and adjacency list - Trees - Cycles - Properties - Paths and connectedness - Sub graphs - Graph Isomorphism - Operations on graphs - Vertex and edge cuts - Vertex and edge connectivity, Spanning Trees, Euler circuits, Hamiltonian graphs.

Probability Theory: Sample Spaces- Events - Axioms – Counting – Conditional Probability and Bayes' Theorem – The Binomial Theorem – Random variable and distributions: Mean and Variance of a Random variable - Binomial-Poisson-Exponential and Normal distributions, Correlation and Regression.

Sampling Distributions and Descriptive Statistics: The Central Limit Theorem, Distributions of the sample mean and the sample variance for a normal population, Sampling distributions (Chi Square, t, F, z). Test of Hypothesis- Testing for Attributes – Mean of Normal Population – One-

tailed and two tailed tests, F-test, and Chi-Square test - Analysis of variance ANOVA – One way and two-way classifications.

References

1.	Kenneth H. Rosen, "Discrete Mathematics and Its Applications", 8 th Edition, McGraw Hill, 2018.
2.	Kolman, Busby and Ross, "Discrete Mathematical Structures", 6 th Edition, PHI, 2013
3.	Sheldon M. Ross, "Introduction to Probability and Statistics for Engineers and Scientists", 6th edition, Academic Press, 2020.

Course Outcomes (CO)

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

CO1	Understand sets, functions and relations and apply these discrete structures in different areas of computing.
CO2	Analyse and apply logical propositions to test the validity of arguments.
CO3	Apply the concepts of trees and graphs to solve real world problems
CO4	Apply the concepts of probability and statistics to solve machine learning and data science problems

Course Code	**:**	CAS713
Course Title	**:**	Computer Organization and Architecture
Type of Course	**:**	PC
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods	**:**	Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the data representation in a digital computer and explain how operations are performed by computer circuits.
CLO2	To understand the organization of computers, performance evaluation of memory and CPU.
CLO3	Study and analyze the modern processor architecture.

Course Content

Number Systems - Binary Arithmetic - Boolean algebra –Karnaugh Map Simplifications– Fixed Point and Floating-point Arithmetic - IEEE 754 Floating-point Standard — Booth's Algorithm - Logic Gates -Combinational Circuits - Sequential Circuits.

Functional units of Computer – Bus Structure - Instruction Set Architecture - Instruction Formats Addressing Modes - Architecture and Instruction Set of 8086 Microprocessor – Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC) Architecture.

Instruction Execution–Phases of Instruction - Control Unit Operation – Control Unit Design: Hardwired, Microprogrammed, and Nano programmed Control Unit Design – Instruction Pipeline - Data Hazards - Control Hazards - Structural Hazards - Techniques for handling Hazards – Arithmetic Pipeline - Pipeline Optimization Techniques.

Memory – Internal Memory – External Memory - Memory Hierarchy– Cache Memory – Cache Performance Improvement – Block Replacement Strategies - Associative Memory – Virtual Memory –Secondary Storage Devices: Disk and Tape Memory.

Accessing I/O Devices – Standard I/O Interfaces - Interface Circuits - Asynchronous Data Transfer - Programmed Input/output (I/O) – Interrupt Driven I/O – Direct Memory Access (DMA) – DMA Controller – Parallel Processor – Multi-core Processor – Mobile Processor – Embedded Processor – AMD Processor - Graphical Processing Unit (GPU) -Tensor Processing Unit (TPU).

References

1.	Chaudhuri, P. Pal, "Computer Organization and Design", India: PHI Learning, 2008.
2.	Stallings, William, "Computer Organization and Architecture: Designing for Performance", Prentice Hall, 2010.
3.	William Stallings, "Computer Organization and Architecture", Global Edition, Pearson Education, 2015.
4.	John Patrick Hayes, "Computer Architecture and Organization", WCB/McGraw-Hill, 1998.
5.	M . Morris Mano, "Digital Logic and Computer Design", Pearson India, 2017.

Course Outcomes (CO)

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

CO1	Understand basic structure and operation of digital computers.
CO2	Understand the design of ALU to perform arithmetic and logic operations on fixed point and floating numbers.
CO3	Understand different types of instructions and addressing modes supported in the instruction set of CPUs
CO4	Understand the design of the control unit.

Course Code	**:**	CAS715
Course Title	**:**	Data Structures and Algorithms
Type of Course	**:**	PC
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce various data structures
CLO2	Design and analysis of algorithms and their applications.

Course Content

Linear data Structures – Arrays, Records, Linked Lists – Singly, Doubly, Circular linked lists - Stacks: operations and applications, representing Stacks - Queues: operations and applications, representing Queues, types: priority queue, Deque, IRD, ORD – Linked Lists and Applications – Hashing.

Non-Linear data Structures - Binary Trees – Binary Tree Representations – Binary tree Traversals – Binary search trees: Definition, operations - B-trees: Definition, operations – Fibonacci heaps: Definition, operations - Graphs – Matrix and list Representations – Graph Traversals - Data structures for disjoint sets.

Algorithms – Definition and Algorithms as a technology – Design and Analysis of Insertion sort, merge sort, heap sort and quicksort – order statistics - Recurrences and Solving recurrences - Asymptotic notations.

Divide-and-Conquer - The maximum-subarray problem, Multiplication of two large integers, Strassen's algorithm for matrix multiplication - Dynamic Programming – Elements - Matrix-chain multiplication, longest common subsequence (LCS), Optimal binary search trees.

Greedy Algorithms – Elements - An activity-selection problem, Huffman codes and Minimum Spanning tree algorithms - Backtracking and Branch-and-Bound strategies with applications – NP concepts - introduction.

References

1.	T .H. Cormen, C.E. Leiserson, R.L. Rivest and C.Stein, "Introduction to Algorithms", 3rd Edition, MIT Press, 2009.
2.	Robert Sedgewick and Kevin Wayne, "Algorithms", 4 th Edition, Addison Wesely, 2011.
3.	Steve S. Skiena, "The Algorithm Design Manual", 2 nd Edition, Springer, 2008
4.	Anany Levitin, "The Design and Analysis of Algorithms", 3 rd Edition, Pearson, 2012.

5.	Aditya Y. Bhargava, “Grokking Algorithms”, Manning Publications, 2016.

Course Outcomes (CO)

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

CO1	Develop own ADTs according to the nature of problems.
CO2	Design algorithms using various strategies.
CO3	Compute time- and space complexities of various algorithms.

Course Code	**:**	CAS 717
Course Title	**:**	Database Management Systems
Type of Course	**:**	PC
Contact Hours	**:**	40
Prerequisites	**:**	-
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To learn different database models and design of databases
CLO2	To study query languages, transaction management, indexing and hashing

Course Content

Introduction to DBMS- Data Models – DBMS Architecture – Data Abstraction - Data Independence - Database Languages - Entity-relationship model - integrity constraints - Conceptual Design with ER Model-EER Model.

Introduction to relational databases – Relational Model – Keys – Relational Algebra – SQL fundamentals – Advanced SQL features – Dependencies – Axioms - Normal forms – Normalization – Decomposition - Dependency preservation - Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions – Query equivalence - Join strategies - Query optimization algorithms- Storage and File Structures: Indices - B + Trees - hashing.

Transaction Concepts – ACID Properties – Schedules – Serializability – Concurrency Control– Need for Concurrency – Locking Protocols – Two Phase Locking – Deadlock – Transaction Recovery – Save Points – Isolation Levels – SQL Facilities for Concurrency and Recovery.

NOSQL Databases: Introduction – CAP Theorem – Document Based systems – Key value Stores – Column Based Systems – Graph Databases- Hbase : data model and implementations – Hbase clients – Hbase examples – Hive – data types and file formats – HiveQL data definition – HiveQL data manipulation – HiveQL Queries.

References

1.	Silberschatz, Korth and Sudarshan, “DataBase System Concepts”, McGraw-Hill, 7th Edition, 2019
2.	C. J. Date, “An Introduction to Database Systems”, 8th Edition, Addison-Wesley, 2003
3.	R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, 7th Edition, Pearson Education/Addison Wesley, 2015.
4.	Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, 4th Edition, McGraw-Hill, 2018.

5.	Michael Kaufmann and Andreas Meier SQL and NoSQL Databases Modeling, Languages, Security and Architectures for Big Data Management, Springer, 2023

Course Outcomes (CO)

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

CO1	Construct SQL Queries using relational algebra
CO2	Design database using ER model and normalize the database
CO3	Construct queries to handle transaction processing and maintain consistency of the database
CO4	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database

Course Code	**:**	CAS719
Course Title	**:**	Operating Systems Fundamentals
Type of Course	**:**	PC
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the design of an operating system and services provided by the OS.
CLO2	To understand what a process is and how processes are synchronized and scheduled.
CLO3	To acquire knowledge on different approaches to memory management
CLO4	To understand the structure and organization of the file system and disk.

Course Content

Operating System concepts - OS Structure – OS Services - System calls – Process management: Process Concept-Operations on process-Cooperating processes- Inter-process communication. Process schedulingScheduling algorithms.

Threads- Multithreading models – Containers - Process synchronization- critical-section – Synchronization hardware – Semaphores – Classic problems of synchronization – critical regions. Deadlocks: Characterization, Prevention, Avoidance, Detection, and Recovery.

Memory Management: Paging, segmentation, Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-mapped files, Kernel memory allocation.

File Systems: File Concepts, Access and Allocation Methods, Free Space Management. Disk Structure, Disk Scheduling and Disk Management.

Protection and security, Distributed Operating Systems – Distributed system structure, Distributed file system, Mobile Operating systems. Case Studies: Linux, Microsoft Windows

References

1.	Abraham Silberschatz, Peter B. Galvin and Greg Gagne, “Operating System Concepts”, 10th edition, John Wiley & Sons Inc., 2018.
2.	Andrew S. Tanenbaum, and Herbert Bos, “Modern Operating Systems”, 5th Edition, Pearson, 2022
3.	William Stallings, “Operating Systems: Internals and Design Principles”, 7th Edition, Prentice Hall, 2011.

Course Outcomes (CO)

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

CO1	Use system calls for managing processes, memory, and the file system.
CO2	Explore the functionalities of distributed and mobile operating systems.
CO3	Explore various types of operating systems including UNIX, Linux, Android, and Windows.

Course Code	**:**	CAS701
Course Title	**:**	Data Structures and Algorithms Lab using C/C++
Type of Course	**:**	ELR
Prerequisites	**:**	-
Contact Hours	**:**	30
Course Assessment Methods		Continuous Assessment, End Assessment

Course Content

Write programs for the following in C/C++ :

a. Matrix implementation
b. Singly Linked List (SLL) implementation
c. Doubly Linked List (DLL) and Circular DLL implementation
d. Implementing Stack and Queue with operations
e. Implementing Binary Search Tree (BST)
f. BFS & DFS implementation
g. Representing polynomials and performing operations
h. Implementing Minimum Spanning Tree (MST) algorithm [Prim's & Kruskal's]
i. Implementing Binary Tree and related operations
j. Tree traversal implementation

Course Outcomes (CO)

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

CO1	Write C/C++ programs for problem solving.
CO2	Implement linear and nonlinear data structures to solve real-time problems.
CO3	Perform searching and sorting techniques on different application domains.
CO4	Implement different algorithm design strategies to solve complex problems.

Course Code	**:**	CAS703
Course Title	**:**	Database Management Systems Lab
Type of Course	**:**	ELR
Prerequisites	**:**	-
Contact Hours	**:**	30
Course Assessment Methods		Continuous Assessment, End Assessment

Course Content

1. Exercise based on SQL for:
   1 Data Definition, Table Creation, Constraints
   2 Insert, Select Commands, Update & Delete Commands.
   3 Inbuilt functions in RDBMS.
   4 Nested Queries & Join Queries.
   5 Set operators & Views.
   6 Control structures.
2. Exercises using Procedures Functions Triggers (PL/SQL) Front End Tool Forms - Menu Design – Reports. Exercises using cursors.
3. Exercises dealing with Integrity Constraints
4. Working with NOSQL
5. Database Design and implementation (for Information Systems)

Course Outcomes (CO)

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

CO1	Construct SQL Queries using relational algebra.
CO2	Design database using ER model and normalize the database.
CO3	Construct queries to handle transaction processing and maintain consistency of the Database.
CO4	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database.

SEMESTER II

Course Code	**:**	CAS712
Course Title	**:**	Networking Technologies
Type of Course	**:**	PC
Prerequisites	**:**	CAS719
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To learn state-of-the-art network reference models and computer network architectures.
CLO2	To understand the internet addressing system, design issues and functionalities of different networking layers.
CLO3	To gain comprehensive knowledge about the standard communication protocols, and recent advancement in networking technologies.

Course Content

Computer network - Network interface card (NIC) – Communication Media - Networking Devices: Hub, Switch, Router and Firewall -Network Protocols – Service Identification - IPv4 Addressing System - Subnetting – Supernetting - IPv6 Addressing System - Circuit Switching -Packet Switching.

Network Topologies -Network Architectures - Open System Interconnect (OSI) Reference Model - TCP/IP Model - TCP Operation - UDP Operation - Congestion control – Flow Control – Error Control.

LAN Components –Packet Switching and Forwarding - LAN Technologies: Ethernet (IEEE 802.3), Token Bus, Token Ring - Wireless LAN (IEEE 802.11) - Virtual LAN (VLAN).

WAN Components - WAN Technologies – WAN Encapsulation - Routed Protocols (IP and IPX) - Static Routing - Dynamic Routing - Routing Protocols - Virtual Private Network (VPN) -Software Defined Networking (SDN).

Point-to-Point Protocol (PPP) - Logical Link Control Protocol (LLC) - Address Resolution Protocol (ARP) -Internet Control Message Protocol (ICMP) - Dynamic Host Configuration Protocol (DHCP) - Domain Name System (DNS) – HTTP, HTTPS - File Transfer Protocol (FTP) - Simple Mail Transfer Protocol (SMTP) - Remote Administration Protocols: Telnet and Secure Shell (SSH).

References

1.	William Stallings, "Data and Computer Communications", 10th Edition, Pearson, 2014.
2.	Stallings, William., and Tanenbaum, Andrew S., "Computer Networks", Pearson Education, Limited, 2009.
3.	Behrouz A. Forouzan, "Data Communications and Networking", 5 th Edition, McGraw-Hill, 2017.
4.	James F. Kurose and Keith W. Ross, "Computer Networking - A Top-Down Approach", 8th Edition, Pearson, 2017.
5.	Andrew S. Tanenbaum, and David J. Wetherall, "Computer Networks", 5th Edition, Pearson, 2011.

Course Outcomes (CO)

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

CO1	Describe available LAN and WAN Technologies.
CO2	Describe the principles of packet switching, forwarding, and routing.
CO3	Distinguish between TCP and UDP packet formats.
CO4	Describe the available application protocols and networking services.

Course Code	**:**	CAS702
Course Title	**:**	Internet Programming Lab
Type of Course	**:**	ELR
Prerequisites	**:**	-
Contact Hours	**:**	30
Course Assessment Methods		Continuous Assessment, End Assessment

Course Content

Networking

1. Learn to use Simulation Tools.
2. Write a HTTP web client program to download a web page.
3. Write applications using TCP sockets and UDP sockets.
4. Simulation of DNS using UDP sockets.
5. Simulation of Distance Vector/ Link State Routing algorithm.
6. Performance evaluation of Routing protocols using Simulation tools.

Web Programming

1. Design web pages using HTML and CSS functionalities.
2. Design and develop client side scripts for Validating Web Form Controls.
3. Develop JavaScript application using event handling.
4. Write programs in Java using Servlets:
    - To invoke servlets from HTML forms.
    - To invoke servlets from Applets.
5. Write programs in Java to create three-tier applications using JSP and Databases for conducting online examinations for displaying student mark lists. Assume that student information is available in a database which has been stored in a database server.
6. Write programs using XML, XSLT/XSL.
7. Develop PHP applications for Session Handling.
8. Design and develop PHP applications connecting with MySQL

At the end of the lab, students will be able to

CO1	Use and explore simulation tools for networking configuration, application, applying algorithms.
CO2	Develop web pages with markups, style sheets, multimedia elements.
CO3	Design and implement real time applications using Javascript.
CO4	Design and implement client server applications using databases, server side technologies

SEMESTER III

CAS749 PROJECT WORK –Phase I

- To explore various research papers pertaining to chosen domain and arrive survey.
- To implement and demonstrate the studies done.

Outcome: To submit papers in conference or journal.

SEMESTER IV

CAS 750 PROJECT WORK –Phase II

- Internal/External project work of 6 Months duration with submission of thesis and viva-voce examination.

Outcome: To submit papers in conference or journal.

ELECTIVES:

Course Code	**:**	CAS7A1
Course Title	**:**	Artificial Intelligence
Type of Course	**:**	PE
Prerequisites	**:**	
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	Compare AI and non-AI solutions.
CLO2	Apply AI techniques in problem-solving.
CLO3	Analyze the best search technique and implement it in real-life applications.
CLO4	Classify supervised and unsupervised learning and knowledge representation.

Course Content

Agents and Environments – Good Behavior: The Concepts of Rationality – The Nature of Environments – The Structure of Agents – AI Solutions Vs. Conventional Solutions is a philosophical approach and a practical approach.

Problem-solving using Search Techniques- Problems- Solutions-Optimality-Uninformed Search Strategies; BFS; DFS; DLS; UCS; IDFS; BDS. Informed Search Strategies-Greedy Best-First-A* Search- Heuristic Functions-Local Search algorithms: Hill Climbing, genetic Algorithms.

Logical Agent – Proposition Logic – Syntax and Semantics – Theorem Proving – Model Checking – Inference in First Order Logic: Forward Chaining – Backward Chaining – Resolution.

Common Sense Vs. Learning-Components- Representations- Forms of learning, Feedback-Learning Types: Supervised; Unsupervised-Reinforcement Learning- Decision trees-Artificial Neural Networks: Introduction, types of networks; Single Layer and Multi-Layer n/w.

Applications of Artificial Intelligence- Natural Language Processing, Speech recognition, Computer vision, Expert systems.

References

1.	Stuart Russell S & Norvig P, Artificial Intelligence: A Modern Approach, 4th Edition, Pearson, 2020
2.	Rich and Knight, Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2014.
3.	Ethem Alpaydin, Machine Learning the New AI, MIT Press, 2016.

4.	Richard E. Neapolitan and Xia Jiang, Artificial Intelligence -With an Introduction to Machine Learning, 2 nd Edition, CRC Press, 2018.

Course Outcomes (CO)

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

CO1	Compare between AI and non-AI solutions.
CO2	Apply AI techniques in problem-solving.
CO3	Analyze the best search technique and implement it in real-life applications.
CO4	Classify supervised and unsupervised learning and knowledge representation.

Course Code	**:**	CAS7A2
Course Title	**:**	Cryptography
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce basic components of information systems security
CLO2	To adopt symmetric ciphers to protect data at rest and in transit.
CLO3	To introduce public key cryptosystem to protect data at rest and in transit
CLO4	To adopt Blockchain-based security system to protect information systems from ever-growing Cyber-attacks.

Course Content

Principles of Information Security - the CIA triad - Security Attacks – Cryptography - Caesar Cipher - Substitution Techniques - Polyalphabetic Substitution Techniques - Cryptographic Key - Transposition Cipher - Symmetric Key vs Asymmetric Key Cryptography - Key Management - Fast Modular Arithmetic - Diffie-Hellman Key Exchange Algorithm - Attack on Diffie-Hellman Key Exchange Algorithm - Cryptanalysis.

Block Ciphers - Design Principles for Block Cipher - Data Encryption Standard (DES) - Triple DES – International Data Encryption Algorithm (IDEA) - Advanced Encryption Standards (AES) - Electronic Code Book (EBC) - Cipher Block Chaining (CBC) - Cipher Block Feedback Mode - Output Feedback Mode - Counter Feedback - Stream Ciphers - Design Principles for Stream Cipher - RC4.

Principles of Public Key Cryptosystems - The RSA Algorithm - Extended Euclidean Algorithm - Key Management - Internet Key Exchange (IKE) - Public Key Infrastructure - Elliptic Curve Arithmetic - Elliptic Curve Cryptography (ECC).
Authentication Requirements - Message Authentication Code (MAC) Hash Functions - Security of Hash Functions and MACs - Secure Hash Algorithm (SHA) - Message Digest 5 (MD5) - Hash-based Message Authentication Code (HMAC) - Cipher-based Message Authentication Code (CMAC) - Digital Signatures - Digital Signature Standard (DSS) - Digital Signature Algorithm (DSA) - Elliptic Curve implementation of DSA (ECDSA) - Kerberos, X.509 Authentication Service - Pretty Good Privacy (PGP) - Privacy Enhanced Mail (PEM).

Blockchain for information security - Homomorphic Encryption - Implications of Quantum Computing on cryptography - Quantum Cryptography - Quantum Key Distribution (QKD) - Post-quantum Cryptography.

References

1.	Deven N. Shah, Mark Stamp's "Information Security Principles and Practices", Wiley India 2010.
2.	Nina Godbole, "Information Systems Security", Willey Student Edition 2009.
3.	Bruce Schneier, "Applied cryptography: protocols, algorithms, and source code in C" (2nd edn.).
4.	Douglas R. Stinson, "Cryptography: theory and practice", CRC Press, 2nd Edition 2005.
5.	Thomas Vidick and Stephanie Wehner, "Introduction to Quantum Cryptography", Cambridge University Press, 2024.

Course Outcomes (CO)

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

CO1	Explain basic components of information systems security
CO2	Design and use symmetric ciphers to protect data at rest and in transit
CO3	Design and use public key cryptosystems to protect data at rest and in transit.
CO4	Design and use Blockchain-based security systems to protect information systems from ever-growing Cyber-attacks.

Course Code	**:**	CAS7A3
Course Title	**:**	Distributed computing
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the concepts and architectures of distributed computing paradigms
CLO2	To illustrate the issues of synchronization, message ordering, and group communication
CLO3	To describe distributed mutual exclusion and distributed deadlock detection techniques
CLO4	To understand distributed shared memory concepts, checkpointing and rollback recovery

Course Content

Introduction- Distributed Systems Architectures - Processes – Communication.

Synchronization - Consistency and Replication - Security - Distributed Object based Systems - Distributed File Systems.

Web Services: SOAP, WSDL, UDDI - Web Services Discovery and Composition - REST based Web Services – Microservices.

Cloud Computing Properties and Characteristics - Cloud Computing Architecture - Cloud Computing Service Delivery Models: Infrastructure as a Service, Platform as a Service, Software as a Service - Deployment Models: Public cloud, Private cloud, Hybrid cloud – Case Studies: Amazon AWS, Microsoft Azure, Amazon EC2, Google Cloud.

Business Drivers for Adopting Cloud Computing - Service Level Agreements– Pricing Models of Cloud - Migrating to Cloud – Task Scheduling - Resource Management - Cloud Security and Privacy - Emerging Trends in Cloud Computing.

References

1.	Andrew S Tanenbaum and Maarten Van Steen, Distributed Systems: Principles and Paradigm, 2015.
2.	Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing Principles, Algorithms, and Systems. Cambridge University Press 2008.

Course Outcomes (CO)

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

CO1	Explain the concepts and technologies of distributed systems.
CO2	Learn the concepts of designing and developing web services.
CO3	Understand cloud computing architecture and techniques.
CO4	Analyze and evaluate cloud computing models.

Course Code	**:**	CAS7A4
Course Title	**:**	Soft Computing Techniques
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce the techniques of soft computing
CLO2	To explain the hybridization of soft computing systems which differ from conventional AI and computing in terms of its tolerance to imprecision and uncertainty.
CLO3	To become familiar with image compression and recognition methods
CLO4	To study the image segmentation and representation techniques

Course Content

Soft Computing and its Techniques, Soft Computing vs Hard Computing. Need for Soft Computing, Applications of Soft Computing in the current industry.

Neural Network (NN), Biological foundation of Neural Network, Neural Model and Network Architectures, Perceptron Learning, Supervised Hebbian Learning, Back-propagation, Associative Learning, Competitive Networks, Hopfield Network, Computing with Neural Nets, and applications of Neural Network.

Genetic Algorithm, Concepts, Operators, Function Optimization, Dominance, Swarm Intelligence, Modeling Collective Behavior in Social Insects, Division of Labor and Task Allocation.

Fuzzy Sets, Operations on Fuzzy sets, Fuzzy Relations, Fuzzy Measures, Defuzzification, Fuzzy Systems, Fuzzy Clustering, Fuzzy Decision Making, Applications of Fuzzy Set Theory to different branches of Science and Engineering.

Neuro Fuzzy and Soft Computing, Adaptive Neuro-Fuzzy Inference System Architecture, Hybrid Learning Algorithm, Learning Methods that Cross-fertilize ANFIS and RBFN, Coactive Neuro Fuzzy Modeling, Framework, Neuron Functions for Adaptive Networks, Neuro Fuzzy Spectrum. Hybridization of other techniques.

References

1.	J. S. R. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing, First Edition, PHI, 2015.
2.	G. J. Klir, and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, First Edition, Prentice- Hall, 1995.

3.	S. Rajasekaran and G. A. V. Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms, PHI, 2017
4.	D. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, 13th Edition, Pearson Education,1989.
5.	E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford Press, 1999

Course Outcomes (CO)

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

CO1	Explain the basics of soft computing and their suitable industry related applications.
CO2	Apply neural network principles and algorithms for given problems
CO3	Apply the principles of fuzzy algorithms for real time applications.

Course Code	**:**	CAS7B1
Course Title	**:**	Digital Image Processing
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To become familiar with digital image fundamentals
CLO2	To get exposed to simple image enhancement techniques in the Spatial and Frequency domain.
CLO3	To learn concepts of degradation function and restoration techniques
CLO4	To study the image segmentation and representation techniques

Course Content

Fundamentals of Image Processing: Introduction to Digital Image Processing - Characteristics of Digital Image - Basic relationship between pixels - Image sampling and quantization – Color models - Basic Geometric Transformations - Fourier Transform - Cosine-Sine and Hartley Transform - Hadamard-HaarSlant Transform - Discrete Fourier Transform.

Image Enhancement in the Spatial and Frequency Domain Filtering: Basic Intensity Transformation Functions - Histogram Processing - Basics of spatial filtering - Smoothing and Sharpening Spatial filters - the basics of filtering in the Frequency Domain - Image smoothing and sharpening using Frequency Domain Filters- Ideal - Butterworth and Gaussian Filters - Homomorphic filtering - Color image enhancement.

Image Restoration: A model of the image Degradation/Restoration process - Noise models - mean filters - inverse filtering -Wiener filtering - Geometric Mean Filter. Image Compression: Fundamentals, Types of redundancies - Lossy and Lossless compression - Entropy of an information source - Shannon Fano Coding - Huffman Coding - Golomb Coding - Arithmetic Coding - LZW coding - Run length coding.

Morphological Image Processing: Basics, Erosion and Dilation, Opening and Closing, The Hit or Miss Transform, Morphological Algorithms-Boundary extraction, Hole filling, convex hull, thinning, skeletons.

Image Segmentation: Fundamentals, Basics of Point, Line, Edge detection, Thresholding, Iterative thresholding, Otsu's method, Multivariable thresholding, Region based segmentation, Segmentation using Morphological Watershed algorithm, The use of motion in segmentation.

References

1.	Rafael C Gonzalez, Richard E Woods, "Digital Image Processing", Fourth Edition, Pearson Education, 2018.
2.	William K Pratt, "Digital Image Processing", Fourth Edition, John Wiley, 2010.
3.	S E Umbaugh, "Digital Image Processing and Analysis: Application with MATLAB and CVIP Tools", Third Edition, Taylor & Francis, CRC Press, 2018.
4.	Frank Y. Shih, "Image Processing and Pattern Recognition", Wiley – IEEE Press, 2010.

Course Outcomes (CO)

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

CO1	Know and understand the basics and fundamentals of digital image processing, such as digitization, sampling, quantization, and 2D-transforms.
CO2	Operate on images using the techniques of smoothing, sharpening and enhancement.
CO3	Understand the restoration concepts and filtering techniques.
CO4	Learn the basics of segmentation, features extraction, compression and recognition methods for color models.

Course Code	**:**	CAS7B2
Course Title	**:**	Theory of Computation
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To learn the principles of what can be computed and how fast
CLO2	To understand the applications of Computation to translators, string searching, and control circuit design.
CLO3	To learn about hierarchy of finite state machines, pushdown machines, context free grammars and Turing machines
CLO4	To learn the notions of decidability, complexity theory, recursive and recursively enumerable languages

Course Content

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) - Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions - Finite Automata and Regular Expressions - Converting from DFA to Regular Expressions - Converting Regular Expressions to Automata - applications of Regular Expressions. REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular - Pumping lemma, applications, Closure properties of regular languages.

CONTEXT FREE GRAMMAR (CFG): Derivation Trees - Sentential Forms - Rightmost and Leftmost derivations of Strings - Ambiguity in CFG's - Minimization of CFG – CNF – GNF - Pumping Lemma for CFLs - Enumeration of Properties of CFL (Proof's omitted).

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL - Acceptance by Final State and Acceptance by Empty stack and its Equivalence - Equivalence of CFG and PDA.TURING MACHINES (TM): Formal definition and behaviour - Languages of a TM - TM as acceptors and TM as a computer of integer functions - Types of TMs.

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL): Properties of recursive and recursively enumerable languages - Universal Turing machine - The Halting

problem - Undecidable problems about TMs - Context sensitive language and linear bounded automata (LBA) - Chomsky hierarchy – Decidability - Post's correspondence problem (PCP) - undecidability of PCP.

References

1.	1. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman (2007), "Introduction to Automata Theory Languages and Computation", 3 rd edition, Pearson Education.
2.	K. L. P Mishra, and N. Chandrashekaran, "Theory of Computer Science- Automata Languages and Computation", 2nd edition, Prentice Hall of India, 2003.
3.	Peter Linz, and Susan H. Rodger, "An Introduction to Formal Languages and Automata", 7th Edition, Jones & Bartlett Learning, 2022.
4.	Gyorgy Revesz (2015), "Introduction to Formal Languages", Dover Publications.
5.	Stefano Crespi Reghizzi, Luca Breveglieri, and Angelo Morzenti, "Formal Languages and Compilation", 3rd Ed., Springer, 2019.

Course Outcomes (CO)

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

CO1	Define languages by abstract, recursive definitions and regular expressions.
CO2	Design a finite automaton to recognize a given regular language.
CO3	Transform a language into regular expression or finite automaton or transition graph.
CO4	Define deterministic finite automata and nondeterministic finite automata.

Course Code	**:**	CAS7B3
Course Title	**:**	Fault Tolerance Systems and Techniques
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the principles of fault-tolerant systems and techniques
CLO2	To understand the reliability and security of software in the sense of fault tolerance.

Course Content

Introduction to Fault-Tolerance: Error - Faults and Failures - Reliability and Availability - Dependability Measures - Mathematical Reliability Modeling - Probability Basics - Reliability and Availability Modeling - Analysis using Markov Models.

Hardware Fault-Tolerance -Canonical and Resilient Structures - Reliability Evaluation Techniques and Models - Processor-level Fault Tolerance - Byzantine Failures and Agreements.

Information Redundancy - Error Detection/Correction Codes (Hamming, Parity, Checksum, Berger, Cyclic, Arithmetic) - Encoding/Decoding circuits - Resilient Disk Systems (RAID) - Fault-Tolerant Networks: Network Topologies and their Resilience - Fault-tolerant Routing.

Software Fault-Tolerance - Single-Version Fault Tolerance - N-Version Programming - Recovery Approach - Exception and Conditional (Assert) Handling - Reliability Models - Check pointing: Optimal Checkpointing - Check pointing in Distributed and Shared-memory Systems.

Fault-Tolerant System Design/Applications - Defect-tolerance in VLSI Designs - Fault Detection in Cryptographic Systems.

References

1.	Israel Koren and C. Mani Krishna, “Fault Tolerant Systems”, 2 nd Edition, 2020.
2.	Elena Dubrova; Fault-Tolerant Design; Springer, 2013.
3.	Michael R. Lyu, “Handbook of Software Reliability Engineering”, IEEE Computer Society Press (and McGraw-Hill), 1996.
4.	Martin L. Shooman, “Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design”, John Wiley & Sons Inc., 2002.
5.	Kishor S. Trivedi, “Probability and Statistics with Reliability, Queuing and Computer Science Applications”, John Wiley & Sons Inc., 2016.

Course Outcomes (CO)

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

CO1	Understand the basic concepts of fault tolerance
CO2	Practice the application principles of the fault tolerance
CO3	Analyse the fault tolerance in hardware and software level

Course Code	**:**	CAS7B4
Course Title	**:**	MOBILE COMPUTING
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To be familiar with mobile technologies, protocols, architectures, and applications.
CLO2	To gain comprehensive knowledge about the state-of–the-art mobile networks.

Course Content

Detailed Introduction of Mobile Computing: History, Types, Benefits, Application, Evolution - Security Concern regarding Mobile Computing - Different Propagation Modes - Wireless Architecture, and its types, needs of mobile user - The cellular concept - Cellular system, hexagonal geometry cell and concept of frequency reuse - Channel Assignment Strategies Distance to frequency reuse ratio.

Telecommunication System: GSM: - Channel allocation, call routing Architecture - PLMN interface - addresses and identifiers - network aspects - frequency allocation - authentication and security - Handoffs Technique. GPRS: network operation, data services, Applications, Billing and charging.

Mobile IP: Need of mobile IP, IP packet delivery, Agent Discovery, Registration, Tunnelling and encapsulation, Route optimization, IP Handoff.

Mobile Transport Layer: Overview of Traditional TCP and implications of mobility control. Improvement of TCP: Indirect TCP, Snoop TCP, Mobile TCP, Fast Retransmit/fast recovery, Time-out freezing, Selective retransmission, Transaction-oriented TCP. Wireless Application Protocol: Introduction of WAP, WAP applications, WAP Architecture, WAP Protocol Stack, Challenges in WAP.

Mobile Ad Hoc wireless networks: Introduction, Benefits, Difference, Routing protocols for ad hoc wireless networks: DSDV and AODV, Introduction to 4G: Introduction, features and challenges, Applications of 4G, 4G network architecture.

References

1.	Asoke K Talukder, and Roopa R Yavagal, “Mobile Computing Technology, Applications and service creation”, TMH, 2017
2.	Raj Kamal, “Mobile Computing”, Oxford, 2018.

3.	William Stallings, “Wireless Communications & Networks”, 2nd Edition Pearson, 2016
4.	Kumkum Garg, “Mobile Computing Theory and Practice”, Pearson, 2010
5.	Behrouz A Forouzan, “TCP/IP Protocol Suite”, 3 rd Edition by TMH, 2008.

Course Outcomes (CO)

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

CO1	Understand concepts of Mobile Communication.
CO2	Analyse next generation Mobile Communication System.
CO3	Understand network and transport layers of Mobile Communication.
CO4	Analyze various protocols of all layers for mobile and ad hoc wireless communication networks.

Course Code	**:**	CAS7C1
Course Title	**:**	GPGPU Programming
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce the features of massively parallel programming architecture.
CLO2	To utilize massively parallel computing capability of a GPU for high performance computing requirements.
CLO3	To provide an overview of parallel design paradigms

Course Content

Introduction: CPU Design – Latency Oriented, GPU Design – Throughput Oriented – Need to use both - Software Cost – Scalability – Portability - GPU Introduction and Architecture - History of GPU Computation - GPGPU Frameworks - Graphics Processor Architecture - Compute Capability - DropIn Libraries - OpenACC Directives.

Parallel Programming Paradigms: Overview, Element Addressing - Multidimensional Kernel – MapGather – Scatter – Reduce – Scan - Thread Handling – Overview - Barrier Synchronization - Thread Synchronization Demo - Warp Divergence - Matrix Multiplication.

CUDA Tools and APIs: Tools Overview, Using NSight Visual Studio and Eclipse - Running CUDA Apps – Debugging – Profiling - CUDA Architecture - CUDA APIs - CUDA 5.5 and 6 Features. CUDA programming: Overview - Compilation Process - Von Neumann Processor and CUDA Thread -Execution Model - first program in CUDA (Vector Addition) - Location Qualifiers - Grid and Block Dimensions - Global Memory - Constant and Texture Memory - Shared Memory - Register and Local Memory - Data Movement - Error Handling - Device Introspection.

Atomics: Overview - Need for Atomics - Atomic Functions - Atomic Sum - Monte Carlo Pi - Handling Events and Streams – Overview – Events - Event API - Event example.

References

1.	Wen-mei Hwu, David Kirk, and Izzat El Hajj, “Programming Massively Parallel Processors – A hands-on approach”, 4th Edition, 2022
2.	Thomas Rauber and Gudula Runger, “Parallel Programming for Multi-core and Cluster Systems”, ACM Computing classification, 1998
3.	Jason Sanders and Edward Kandrot, “CUDA by Example”, Addison Wesley, 2010
4.	Shane Cook, “CUDA Programming - A Developer’s Guide to Parallel Computing with GPUs”, Morgan Kaufmann Publishers, 2012

Course Outcomes (CO)

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

CO1	Analyze GPU Design for finding solutions.
CO2	Work with parallel programming paradigms.
CO3	Solve high performance computing problems using GPUs.

Course Code	**:**	CAS7C2
Course Title	**:**	Software Engineering
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To impart concepts of a comprehensive study on the theories, processes, methods, and techniques of building high-quality software in cost-effective ways

Course Content

Introduction to Software Engineering, Software Life Cycle Models, Requirements Analysis and Specification: formal requirements specification and verification - axiomatic and algebraic specifications.

Software Design Issues, Function Oriented Software Design, Object Modeling using UML: use case model, class and interaction diagrams, activity and statechart Diagrams, Object Oriented Software Development: design patterns, domain modeling, User Interface Design.

Coding and Testing: code review, black box testing, white box testing, debugging, integration and system testing, Automation testing tools - Software Maintenance, Software Reuse.

Software Project Planning: Project planning and estimation, cost and staffing level estimation, Software Project Monitoring and Control, Software Reliability and Quality Management, Risk Management and Software Quality Assurance.

Agile Software Development - Agile Manifesto and Principles – Agile Project Management (Lean Software Management and DevOps) – Agile and Lean Frameworks: SCRUM, Crystal, Kanban, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview – lifecycle – roles, practices and Applicability.

References

1.	Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2017
2.	Roger Pressman, Software Engineering: A Practitioner's Approach, 8th Edition, McGraw Hill, 2014
3.	Rajib Mall, Fundamentals of Software Engineering, 5th Edition, PHI Learning, 2018
4.	Craig Larman, Agile and Iterative Development: A Manager's Guide, 1st Edition, Addison Wesley
5.	David J. Anderson, Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Prentice Hall, 2003

Course Outcomes (CO)

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

CO1	Demonstrate a basic understanding of software engineering practices from vision to analysis, design, development, validation, deployment and maintenance.
CO2	Develop skills to create and use various software Engineering based techniques and tools to solve real world problem.
CO3	Estimate cost, effort and risk involved in a software project development

Course Code	**:**	CAS7C3
Course Title	**:**	Computer Vision and Pattern Recognition
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the fundamental concepts related to image processing, feature extraction, and pattern analysis
CLO2	To apply the concepts to solve computer vision problems of different fields

Course Content

Fundamentals of Image Formation - Transformation: Orthogonal – Euclidean - Affine-Projective, etc; Fourier Transform - Convolution and Filtering - Image Enhancement-Histogram Processing.

Edges - Canny, LOG, DOG; Line detectors (Hough Transform)- Corners - Harris and Hessian Affine, Orientation Histogram - SIFT, SURF, HOG, GLOH - Scale-Space Analysis - Image Pyramids and Gaussian derivative filters - Gabor Filters and DWT.

Image Segmentation: Region Growing - Edge Based approaches to segmentation - Graph-Cut - MeanShift – MRFs -Texture Segmentation; Object detection.

Motion analysis: Background Subtraction and Modeling - Optical Flow – KLT - Spatio-Temporal Analysis Dynamic Stereo; Motion parameter estimation.

Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians, - Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised - Classifiers: Bayes, KNN, ANN models - Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

References

1.	Richard Szeliski, “Computer Vision: Algorithms and Applications”, 2 nd Edition, Springer, 2022.
2.	D. A. Forsyth, and J. Ponce, “Computer Vision: A Modern Approach”, Pearson Education, 2003
3.	Mark S. Nixon and Alberto S. Aguado, “Feature Extraction & Image Processing for Computer Vision”, 3rd Edition, Academic Press, Elsevier, 2012
4.	R. Gonzalez and R. E. Wood, “Digital Image Processing”, 3 rd Edition, Prentice Hall of India, 2008
5.	K.Pratt, “Digital Image Processing”, 4 th Edition, McGraw Hill, 2007

6.	R. O. Duda, P. E. Hart and D. Stork, “Pattern Classification” 2 nd Edition, Wiley, 2002
7.	C. Bishop, “Pattern Recognition and Machine Learning”, Springer 2006

Course Outcomes (CO)

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

CO1	Apply fundamental algorithms in Image Processing and analyze their applicability for real time problems.
CO2	Design solutions for various computer vision and pattern recognition problems
CO3	Design and develop innovative computer vision applications or systems.

Course Code	**:**	CAS7C4
Course Title	**:**	Web Applications Development
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	50
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To comprehend the basics of the internet and web terminologies.
CLO2	To introduce scripting language concepts for developing client-side applications.
CLO3	To practice server-side programming features – ASP .NET, PHP, JSP
CLO4	To be familiar with database applications

Course Content

Internet and World Wide Web: Introduction to Internet, www, Internet browsers Netscape & Explorer, Introduction to Client Server Architecture/Computing, History of the web, Growth of the web, Protocols governing the web, resources of Internet, H/W & S/W requirements of Internet, Internet service providers, Internet Services, Internet Clients, and Internet Servers. Concept of E- Commerce and E-governance.

Markup Languages: Introduction to HTML, Formatting Tags, Links, Lists, Tables, Frames, Forms, Comments in HTML, DHTML and XML Documents, Data Interchange with an XML document, Document type definition, Features and Applications, Working with Style sheets.

Client-side Scripting: Scripting basics, Introducing JavaScript, Documents, Statements, Functions, Objects in Javascript, Events and Event handling, Arrays, Forms, Buttons, Checkboxes, Text Fields, and Text Area – Applications using Javascript – Extensions of Javascript.

Server-side Scripting: Introduction to server-side scripting language, RMI, The Problem with Servlet. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat- Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging - Sharing Data Between JSP pages, Requests and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations – Applications using server side scripting.
PHP: Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP myadmin and database bugs – Applications using PHP - Django Framework.

References

1.	Robin Nixon, "Learning PHP, MySQL and Javascript", 6 th Edition, O'Reilly, 2021
2.	Laura Lemay, Rafe Colburn, and Jennifer Kyrnin, "Mastering HTML, CSS and JavaScript Web Publishing", BpB Publishers, 2016
3.	Casimir Saternos, "Client-Server Web Apps with JavaScript and Java", Shroff Publishers, 2014
4.	Nicholas S. Williams, "Professional Java for Web Applications", Wiley, 2014.
5.	Paul Deitel, Harvey Deitel, and Abbey Deitel, "Internet and WWW", Pearson Education, 2018

Course Outcomes (CO)

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

C01	Understand significance and features of Web Applications.
C02	Apply Markups, Style Sheets, XML technologies in designing websites and developing web applications.
C03	Design and develop client side applications using javascript for real time scenarios.
C04	Build and deploy client-server applications using databases and server side technologies.

Course Code	**:**	CAS7D1
Course Title	**:**	Defensive and Secure Software Development
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To learn software security fundamentals.
CLO2	To understand software development with security built in.
CLO3	To design and develop security-driven design software.

Course Content

Introduction - Types of security; Correctness vs. security; Assets and threats; Security mindset; Layers of security – Software security - Why it is important; Security in the Software Development Lifecycle (SDLC); Kinds of attacks; How to design security in; Defense in depth and breadth; Principles, tactics, guidelines, and standards.

Programming Language Overviews - Overviews of secure software of JavaScript, Java, C, and C++ - Security in design - Domain-driven design; Requirements definitions; Threat models - Security in Programming - Defensive coding; Immutability; Design by contract; Validation (origin, size, bounds, lexical, syntactic, and semantic checking); Integrity maintenance (defensive copying, failing fast, managing exceptions); Secure error handling; Mitigating data leaks; Reducing complexity.

Security in testing and operations - Types of testing (including fuzz testing and pen testing); SecOps; Continuous integration – Language dependent guidelines and standards for secure programming in C, C++, Java, and JavaScript – Cryptology - math and the algorithms behind cryptography, cryptanalysis, encryption, and hashing.

Authentication and Authorization - difference between Authentication and Authorization; Passwords, hashing and salting; tokens; Defining a good permission system; Digital signatures; SSO; SAML and OAuth.

Web security - OWASP - CWE and CVE - compare Firebase with systems that use a traditional backend - Network Security: Internet architecture and security issues with TCP/IP; what is transport-layer security and how does it work; AWS IAM – Privacy - Privacy vs. security; Expectations.

References

1.	Dan Bergh Johnsson, Daniel Deogun, and Daniel Sawano, “Secure by Design”, Manning, 2019.
2.	Gary McGraw, “Software Security: Building Security” Addison-Wesley, 2006.
3.	Van Wyk, Kenneth R., Graff, and Mark G., “Secure Coding: Principles and Practices”. N.p.: Turtleback, 2003.
4.	Seacord, Robert C., “Secure Coding in C and C++”, Pearson Education, 2005.
5.	Long, Fred. “The CERT Oracle Secure Coding Standard for Java”, Addison-Wesley, 2012.

Course Outcomes (CO)

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

CO1	Understand software security fundamentals.
CO2	Develop secure software.
CO3	Use secure coding practices in software development.

Course Code	**:**	CAS7D2
Course Title	**:**	Software Verification and Validation
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the proof methods for software verification
CLO2	To write human-readable, machine-checkable proofs of software correctness.
CLO3	To design and develop reusable formal proofs, e.g., of safety/liveness for concurrent programs

Course Content

Introduction - Software engineering - Software Development Life Cycles Models, Conventional Software Life Cycle Models- Object Orientation - Objects and Classes, Features, Object Oriented Software Life Cycle Models, Object oriented Methodologies, Object – Oriented Modeling – Terminologies.

Introduction to software testing techniques, Software Implementation - Quality and Metrics - Software Implementation – Tools and Techniques - Software quality – Software quality models - Measurement basic - analyzing the metric data - Metrics for measuring size and structure – Measuring software quality object-oriented metrics – Overview of Scala for Implementation.

Introduction to Software testing strategies Software Testing Fundamentals: Testing Objectives, Testing Principles, Testability, Test Case Design, White-Box Testing, Basis Path Testing: Flow Graph Notation, Cyclomatic Complexity, Graph Matrices, Control Structure Testing: Condition Testing, Data Flow Testing, Loop Testing, Black-Box Testing - GraphBased Testing Methods, Equivalence Partitioning, Boundary Value Analysis, Comparison Testing, Orthogonal Array Testing, Testing for Specialized Environments, Architectures, and Applications: Testing GUIs, Testing of Client/Server Architectures, Testing Documentation and Help Facilities, Testing for Real-Time Systems.

A Strategic Approach to Software Testing: Verification and Validation, Organizing for Software Testing, A Software Testing Strategy, Criteria for Completion of Testing, Strategic Issues, Unit Testing: Unit Test Considerations, Unit Test Procedures, Integration Testing: Top Down Integration, Bottom-up Integration, Regression Testing, Smoke Testing, Comments on Integration Testing, Integration Test Documentation, Validation Testing: Validation Test Criteria, Configuration Review, Alpha and Beta Testing, System Testing: Recovery Testing, Security Testing, Stress Testing, Performance Testing, The Art of Debugging: The Debugging Process, Psychological Considerations, Debugging Approaches.

Software Quality: McCall's Quality Factors, FURPS, ISO 9126 Quality Factors, The Transition to a Quantitative View, A Framework for Technical Software Metrics: The Challenge of Technical Metrics, Measurement Principles, The Attributes of Effective Software Metrics, Metrics for the Analysis Model: Function-Based Metrics, The Bang Metric, Metrics for Specification Quality, Metrics for the Design Model: Architectural Design Metrics, Component-Level Design Metrics, Interface Design Metrics, Metrics for Source Code, Metrics.

References

1.	Yogesh Singh, and Ruchika Malhotra, "Object-Oriented Software Engineering", PHI, 2012
2.	Timothy C. Lethbridge and Robert Laganiere, "Object-Oriented Software Engineering", McGraw-Hill, 2nd ed., 2004
3.	G. Booch, Benjamin/Cummings, "Object-Oriented Analysis and Design with Applications", 3 rd Edition, Addison-Wesley, 2007
4.	Roger Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill Higher Education, 2010
5.	S. Kenneth Rubin, "Essential Scrum: A Practical Guide to the Most Popular Agile Process", Pearson Publication, 2012
6.	Jason Swartz, "Learning Scala Practical Functional Programming for the JVM", O'Reilly Media, December 2014
7.	Stephen R. Schoch, "Object-Oriented and Classical Software Engineering", Mc Graw Hill, 8th Edition, 2020
8.	Roger Y. Lee, "Object-Oriented Software Engineering With UML, A Hands-On Approach", Nova Publishers, January 2019

Course Outcomes (CO)

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

CO1	Understand the basic concepts of software engineering.
CO2	Practice the application principles of Software testing.
CO3	Analyze the process of software verification and validation

Course Code	**:**	CAS7D3
Course Title	**:**	Machine Learning and Deep Learning Techniques
Type of Course	**:**	PE
Prerequisites	**:**	CAS711
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce the basic concepts and techniques of machine learning.
CLO2	To develop the skills in using recent machine learning techniques for solving practical problems.
CLO3	To be familiar with a set of well-known supervised, unsupervised and reinforcement learning algorithms.
CLO4	To introduce the techniques of deep learning.

Course Content

Introduction: Learning, Designing a Learning System, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Machine Learning workflow, Machine Learning issues and challenges, Introduction to Deep Learning, Machine Learning and Deep Learning Applications.

Supervised Learning: Predictive Models: Regression, Multivariate Regression, Types of Regression Models, Estimation of Regression coefficients, issues and challenges, applications. Classification Models: Introduction, Different types of classifiers, issues and challenges of single classifiers, applications. Unsupervised Learning – Clustering, Mixture Models and EM Algorithm, Fuzzy kMeans Algorithm, applications.

Ensemble Learning: Boosting, AdaBoost Algorithm, Bagging, Random Forest, No-Free-Lunch Theorem, XGBoost Algorithm, Ensemble Diversity, Error Decomposition, Diversity Measures, Evaluating Ensembles of Classifiers.

Reinforcement Learning: Introduction to Reinforcement Learning, Learning Task, Example of Reinforcement Learning in Practice, Learning Models for Reinforcement – (Markov Decision process, Q Learning - Q Learning function, Q Learning Algorithm, SARSA algorithm), Nondeterministic Rewards and Actions, Application of Reinforcement Learning.

Deep Learning: Introduction, Deep Feedforward Networks, Architecture Design; Convolutional Networks – Introduction, Convolution (1D and 2D), Pooling, Training of network, Case study of CNN (Healthcare, Agriculture, Stock Market, Weather Forecasting, etc.). Sequence Modeling: Recurrent Neural Network (RNN) Model, Types of RNNs, Vanishing Gradients with RNN, Gated Recurrent Unit, Long Short-Term Memory (LSTM), Deep Recurrent Neural Networks, RNN for Time Series, Transformer Network Models. Case studies on recent real-world problems.

References

1.	Ethem Alpaydin, Introduction to Machine Learning, 4th Edition, MIT Press 2020.
2.	Tom M. Mitchell, Machine Learning, 1st Edition, McGraw-Hill Education (India) Private Limited, 2017
3.	Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press 2016
4.	Stephen Marsland, Machine Learning: An Algorithmic Perspective, 2nd Edition, CRC Press, 2014
5.	Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag, 2013

Course Outcomes (CO)

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

CO1	Select real-world applications that need machine learning based solutions.
CO2	Implement and apply machine learning algorithms.
CO3	Select appropriate algorithms for solving a particular group of real-world problems.
CO4	Recognize the characteristics of machine learning techniques that are useful to solve real-world problems.

Course Code	**:**	CAS7D4
Course Title	**:**	Compiler Design
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To learn various phases of compiler
CLO2	To learn various parsing techniques
CLO3	To understand intermediate code generation techniques and run-time environment
CLO4	To learn various code optimization techniques

Course Content

Introduction To Compiler - Language implementation methods – Structure of a compiler – Compiler writing tools - Lexical analysis – Role of Lexical Analyzer – Input buffering – Specification of tokens– Recognition of tokens – Regular language - Finite automata - Regular expression - From regular expression to finite automata - Scanner generator (lex,flex).

Syntax Analysis - General problem of describing syntax - Formal methods of describing syntax – Context-Free grammar -The parsing problem - Role of parsers –Top-Down parsers: LL(1) parser – Bottom-Up Parsers: LR(0), SLR(1), CLR(1), LALR(1), Operator precedence parser – Recursive-Descent parser - Parser generator (yacc,bison).

Semantic Analysis and Run Time Environment - Attribute grammar -Syntax directed definition - Evaluation and flow of attribute in a syntax tree - Symbol table - Symbol attributes and management - Procedure activation - Parameter passing - Value return - Memory allocation – Variable scope.

Intermediate Code Generation - Different types of intermediate forms - Intermediate languages – Translation of different language features: declarations – assignment Statements – Boolean expressions – Control statements.

Target Code Generation And Code Optimization - Issues in code generation – Register allocation - Target code generation - Design of simple code generator –Machine-dependent code optimization, Machine-independent code optimization - Analysis: control-flow, and dataflow dependencies- Code improvement local optimization - Global optimization -Loop optimization - Peep-hole optimization - Architecture dependent code improvement: instruction scheduling (for pipeline), and loop optimization (for cache memory).

References

1.	Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, “Compilers: Principles, Techniques, and Tools”, Pearson Education, 2013
2.	Dhamdhere D.M., “Compiler Construction: Theory and Practice”, McMillan India Ltd., 1983
3.	V. Raghavan, “Principles of Compiler Design”, Tata McGraw Hill Education Publishers, 2017
4.	Jean Paul Tremblay, and Paul G Serenson, “The Theory and Practice of Compiler Writing”, BS Publications, 2005
5.	Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thompson Learning, 2003
6.	Allen I. Holub, “Compiler Design in C”, Prentice Hall of India, 2003
7.	C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, Benjamin Cummings, 2003

Course Outcomes (CO)

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

CO1	Explain various phases of the compiler.
CO2	Explain various parsing techniques.
CO3	Explain the process of intermediate code generation.
CO4	Explain various code optimization techniques.

Course Code	**:**	CAS7E1
Course Title	**:**	Big Data Analytics
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce big data analytics and to understand the importance of big data.
CLO2	To introduce different approaches of exploiting big data sources such as social media, mobile devices, and sensors
CLO3	To understand methodologies of analyzing big data.
CLO4	To acquire knowledge of handling unstructured and semi-structured data using NoSQL database

Course Content

BIG DATA OVERVIEW – Classification of Digital Data, Big Data, Evolution of Big Data, Structuring Big Data, Elements of Big Data, Challenges with Big Data, Big Data Analytics, Future of Big Data, Big Data Use Cases (Social Network, Fraud Detection and Prevention, Retail Industry, Healthcare, etc.), Technologies for Big Data.

HADOOP – Hadoop Ecosystem, Hadoop Distributed File System, Components of Hadoop, Analysing the Data with Hadoop, Scaling Out, Hadoop Streaming, HDFS Design; MapReduce – MapReduce Framework, Techniques to Optimize MapReduce Jobs, Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Developing MapReduce Application.

NoSQL DATABASES – NoSQL, Characteristics of NoSQL, History of NoSQL, Types of NoSQL Data Models, CAP Theorem, MongoDB, Neo4j.

FRAMEWORKS – Hive Services, Data Types in Hive, Built-in Functions in Hive, Hive QL; The Pig architecture, Benefits and Properties of Pig, Pig Latin, Pig Operators; HBase, ZooKeeper, Mahout.

DATA VISUALIZATION – Representing Visual Data, Techniques for Visual Data Representation, Types of Data Visualization, Applications of Data Visualization, Visualizing Big Data, Data Visualization Tools, Data Visualization with Tableau.

References

1.	Tom White, “Hadoop: The Definitive Guide”, O’reilly Media, 4th Edition, 2015
2.	Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, and Paul Zikopoulos, “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”, McGraw Hill Publishing, 2015
3.	Jure Leskove, Anand Rajaraman, and Jeffrey D. Ullman, “Mining of Massive Datasets”, Cambridge University Press, 3rd Edition, 2020
4.	Guy Harrison, “Next Generation Databases: NoSQL and Big Data”, APress, 1st Edition, 2015.
5.	Paul Zikopoulos, Dirkde Roos, Krishnan Parasuraman, Thomas Deutsch , James Giles, and David Corrigan, “Harness the Power of Big Data The IBM Big Data Platform”, Tata McGraw Hill Publications, 2012
6.	DT Editorial Services, “Big Data Black Book: (Covers Hadoop 2, MapReduce, Hive, YARN, Pig, R and Data Visualization)”, India: Dreamtech Press, 2016

Course Outcomes (CO)

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

CO1	Comprehend the concepts of big data analytics.
CO2	Build web-intelligence applications exploiting big data using big data platforms based on the ‘map-reduce’ parallel programming framework.
CO3	Effectively use NoSQL databases for storage and retrieval of big data.

Course Code	**:**	CAS7E2
Course Title	**:**	Quantum Computing
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To introduce the fundamentals of quantum computing
CLO2	The problem-solving approach using finite dimensional mathematics
CLO3	Design and create quantum circuits for quantum algorithms to run on quantum computers

Course Content

Vector Spaces, Linear Combination Of Vectors, Uniqueness of a spanning set, basis & dimensions, inner Products, orthonormality, gram-schmidt orthogonalization, braket formalism, the Cauchy-schwarez and triangle, Complex vector spaces, Tensor products of vector spaces, inner products and Hilbert spaces, Hermitian and unitary matrices.

Observables, The Pauli Operators, Pauli Matrix, Hermitian unitary and normal operator, Eigen values & Eigen Vectors, Spectral Decomposition, Trace of an operator, important properties of Trace, Expectation Value of Operator, Projection Operator, Positive Operators, Commutator Algebra, Heisenberg uncertainty principle, polar decomposition & singular values, Postulates of Quantum Mechanics.

Superposition of states, Polarization of light, entanglement Bits and Qubits, Bloch sphere representation of a qubit, multiple qubits. Qubit operations, Hadamard Gate, CNOT Gate, Phase Gate, Z-Y decomposition, Quantum Circuit Composition, Basic Quantum circuits, Distinguishing Quantum states & Measures, Projective Measurements, Measurement on Composite systems, Generalized Measurements, Positive Operator-Valued Measures, Graph states and codes, Quantum error correction.

Quantum Algorithms: Quantum parallelism, Quantum Evolution, Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Simon's periodicity algorithm, Grover's search algorithm, Period finding, Shor's Factoring algorithm. Application of entanglement, teleportation, superdence coding.

Types of errors, Device level metrics, System level metrics, Quantum programming languages, Probabilistic and Quantum computations, introduction to quantum cryptography and quantum information theory.

References

1.	Quantum computing explained, David McMahon, Wiley-interscience, John Wiley & Sons, 2008
2.	Quantum computing for computer scientists, Noson S. Yanofsky, Mirco A. Mannucci, Cambridge University Press 2008
3.	Quantum Computing: A Gentle Introduction (Scientific and Engineering Computation), Eleanor G. Rieffel, Wolfgang H. Polak, MIT Press, 2014
4.	Quantum computation and quantum information, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press 2010
5.	Introduction to Quantum Mechanics, 2nd Edition, David J. Griffiths, Prentice Hall New Jersey 1995

Course Outcomes (CO)

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

CO1	Basics of complex vector spaces
CO2	Quantum mechanics as applied in Quantum computing
CO3	Architecture and algorithms
CO4	Fundamentals of Quantum computations.

Course Code	**:**	CAS7E3
Course Title	**:**	Mobile Applications Development
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	Understand system requirements for mobile applications.
CLO2	Explore and implement suitable design using specific mobile development frameworks

Course Content

Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications.

Basic Design - Introduction – Basics of embedded systems design – Embedded OS - Design constraints for mobile applications, both hardware and software related – Architecting mobile applications – user interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability, and modifiability.

Advanced Design - Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications.

Android - Introduction – Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server-side applications– Using Google Maps, GPS and WiFi – Integration with social media applications.

IOS - Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using WiFi - iPhone marketplace.

References

1.	Jeff McWherter and Scott Gowell, “Professional Mobile Application Development”, Wrox, 2012
2.	Charlie Collins, Michael Galpin and Matthias Kappler, “Android in Practice”, DreamTech, 2012
3.	James Dovey and Ash Furrow, “Beginning Objective C”, Apress, 2012

4.	David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, “Beginning iOS 6 Development: Exploring the iOS SDK”, Apress, 2013 Q

Course Outcomes (CO)

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

CO1	Describe the requirements for mobile applications.
CO2	Develop design for mobile applications for specific requirements.
CO3	Implement the design using mobile application development frameworks.

Course Code	**:**	CAS7E4
Course Title	**:**	Augmented Reality and Virtual Reality
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To explore the concepts and applications of Augmented Reality and Virtual Reality (AR and VR).

Course Content

Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality.

Multiple Models of Input and Output Interface in Virtual Reality: Input -- Tracker, Sensor, Digital Glove, Movement Capture, Video-based Input, 3D Menus & 3D Scanner etc. Output -- Visual /Auditory / Haptic Devices.

Visual Computation in Virtual Reality: Fundamentals of Computer Graphics. Software and Hardware Technology on Stereoscopic Display. Advanced Techniques in CG: Management of Large-Scale Environments & Real Time Rendering, Interactive Techniques in Virtual Reality: Body Track, Hand Gesture, 3D Manus, Object Grasp.

Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc, Application of VR in Digital Entertainment: VR Technology in Film & TV, Production. VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR.

Augmented and Mixed Reality, Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems.

References

1.	Burdea, G. C. and P. Coffet. "Virtual Reality Technology", 2nd Edition. Wiley-IEEE Press, 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.	Erin Pangilinan, Steve Lukas and Vasanth Mohan, "Creating Augmented and Virtual Realities", O'Reilly, 2019.

Course Outcomes (CO)

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

CO1	Understand the principles and use cases of AR and VR.
CO2	Understand various I/O interfaces – functionalities and usage.
CO3	Explore the visual computations present in AR and VR applications.
CO4	Gain knowledge over various tools and frameworks for AR and VR applications development

Course Code	**:**	CAS7F1
Course Title	**:**	Blockchain Technologies
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand blockchain systems (Bitcoin and Ethereum)
CLO2	To design, build, and deploy smart contracts and distributed applications
CLO3	To integrate ideas from blockchain technology into projects.

Course Content

Distributed Database – Byzantine General problem and Fault Tolerance – Hadoop Distributed File System – Distributed Hash Table – ASIC resistance – Turing Complete – Cryptography: – Hash function – Digital Signature - ECDSA – Memory Hard Algorithm – Zero Knowledge Proof.

Blockchain Introduction – Advantage over conventional distributed database – Blockchain Network – Mining Mechanism – Distributed Consensus – Merkle Patricia Tree – Gas Limit – Transactions and Fee – Anonymity – Reward – Chain Policy – Life of Blockchain application – Soft & Hard Fork – Private and Public blockchain.

Distributed Consensus – Nakamoto consensus – Proof of Work – Proof of Stake – Proof of Burn – Difficulty Level – Sybil Attack – Energy utilization and alternate.

Cryptocurrency – Distributed Ledger – Bitcoin protocols – Mining strategy and rewards – Ethereum – Construction – DAO – Smart Contract – GHOST – Vulnerability – Attacks – Sidechain – Namecoin.

Cryptocurrency Regulation – Stakeholders – Roots of Bit coin – Legal Aspects-Crypto currency Exchange – Black Market and Global Economy – Applications: – Internet of Things – Medical Record Management System – Domain Name Service and future of Blockchain.

References

1.	Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (2016)
2.	Andreas Antonopoulos, Satoshi Nakamoto, “Mastering Bitcoin”, 2nd O’Reilly, 2017

Course Outcomes (CO)

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

CO1	Explain design principles of Bitcoin and Ethereum.
CO2	Explain the Simplified Payment Verification protocol.
CO3	Interact with a blockchain system by sending and reading transactions.
CO4	Design, build, and deploy a distributed application.

Course Code	**:**	CAS7E2
Course Title	**:**	Design Patterns
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To comprehend the rationale and benefits of software design patterns.
CLO2	To impart knowledge on the development of good design patterns.

Course Content

Introduction: Introduction to Design Patterns, Object Oriented Analysis and Design, Types of Design Patterns, Applications of Design Patterns, Anti Patterns, Code Refactoring Techniques for design patterns.

Creational Patterns: Factory Methods, Static Factory Pattern, Singleton Pattern, Abstract Factory Pattern, Object Pool Pattern, Prototype Pattern, Builder Pattern, Telescopic Constructor Pattern.

Structural Patterns: Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Façade Pattern, Flyweight Pattern, Private Class Data, Proxy Pattern.

Behavioral Design Patterns - I: Chain of responsibility Pattern, Command Pattern, Interpreter Pattern, Iterator Pattern, Mediator Pattern.

Behavioral Design Patterns – II: Memento Pattern, Null Object Pattern, Observer Pattern, State Pattern, Strategy Pattern, Template method, Visitor Pattern.

References

1.	Johnson, Ralph E.., Gamma, Erich., Vlissides, John., and Helm, Richard, “Design Patterns: Elements of Reusable Object-oriented Software”. Pearson Education, 1995
2.	Robson, Elisabeth., and Freeman, Eric, “Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software”, O'Reilly, 2021
3.	Stephen Stelting and Olav Maassen, “Applied Java Patterns”, Prentice Hall, 2002.
4.	James W. Cooper, “Java Design Patterns - A Tutorial”, Addison-Wesley, 2000.
5.	Joshua Kerievsky, “Refactoring to Patterns”, Addison-Wesley, 2005.

Course Outcomes (CO)

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

CO1	Solve common problems in software design with ease.
CO2	Analyze object-oriented design for patterns.
CO3	Represent design decisions more effectively with examples and architectural use cases.

Course Code	**:**	CAS7E3
Course Title	**:**	Internet of Things
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To understand the fundamentals of Internet of Things (IoT)
CLO2	To study the design and working of IoT devices.
CLO3	To acquire skills to program the IoT devices
CLO4	To develop applications using IoT Devices.

Course Content

IoT Fundamentals, IoT Architecture and Protocols, IoT Platforms, IoT Components and Communication Technologies, IoT Sensors: Temperature, Moisture, Light, Acoustic & Noise, Water level, Presence & Proximity, Motion, Gyroscope, Chemical, Image; IoT actuators, IoT Examples, IoT Challenges.

IoT Protocols - Protocol Standardization – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Unified Data Standards – IEEE 802.15.4 – BACNet Protocol – Modbus– Zigbee Architecture – Network layer –LowPAN - CoAP – Security.

Arduino Uno Architecture, Arduino Programming, IDE Setup, Writing Arduino Software, Arduino Libraries, Basics of Embedded C programming for Arduino, Integration of Sensors and Actuators with Arduino, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi.

Basics of Wireless Networking, Introduction to ESP8266 Wi-Fi Module, Wi-Fi libraries, Web server introduction, installation, configuration, Posting sensor(s) data to web server, IoT Cloud platforms, ThingSpeak API and MQTT, Interfacing ESP8266 with Web services.

IoT Case Studies and Real-World Applications: Smart Cities and Smart Homes, Connected Vehicles, Smart Grid, Smart Farming, Industrial IoT, SCADA Networks, Medical IoT, Activity Monitoring.

References

1.	Haider Raad, "Fundamentals of IoT and Wearable Technology Design", Wiley- IEEE Press, ISBN: 978-1-119-61753-2, 2021.
2.	Adrian McEwen and Hakim Cassimally, "Designing the Internet of Things", John Wiley & Sons, 2013
3.	Cuno Pfister, "Getting Started with the Internet of Things: Connecting Sensors and Microcontrollers to the Cloud", Maker Media, 2011
4.	S. Velliangiri, Sathish A.P. Kumar, and P. Karthikeyan, "Internet of Things: Integration and Security Challenges", CRC Press - 1 st edition 2020

Course Outcomes (CO)

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

CO1	To understand the working of IoT devices.
CO2	To Program the IoT devices.
CO3	To develop applications using IoT devices.

Course Code	**:**	CAS7E4
Course Title	**:**	Real Time Systems
Type of Course	**:**	PE
Prerequisites	**:**	-
Contact Hours	**:**	40
Course Assessment Methods		Continuous Assessment, End Assessment

Course Learning Objectives (CLO)

CLO1	To study issues to the design and analysis of systems with real-time constraints.
CLO2	To learn the features of Real time OS.
CLO3	To study the various Uniprocessor and Multiprocessor scheduling mechanisms
CLO4	To learn about various real time communication protocols

Course Content

Introduction to real time computing - Concepts; Example of real-time applications – Structure of a real time system – Characterization of real time systems and tasks - Hard and Soft timing constraints - Design Challenges - Performance metrics - Prediction of Execution Time : Source code analysis, Microarchitecture level analysis, Cache and pipeline issues- Programming Languages for Real-Time Systems.

Real time OS – Threads and Tasks – Structure of Microkernel – Time services – Scheduling Mechanisms Communication and Synchronization – Event Notification and Software interrupt.

Task assignment and Scheduling - Task allocation algorithms - Single-processor and Multiprocessor task scheduling - Clock-driven and priority-based scheduling algorithms- Fault tolerant scheduling.

Real Time Communication -Network topologies and architecture issues – protocols – contention based, token based, polled bus, deadline based protocol, Fault tolerant routing. RTP and RTCP.

Real time Databases – Transaction priorities – Concurrency control issues – Disk scheduling algorithms – Two phase approach to improve predictability.

References

1.	C.M. Krishna, Kang G. Shin" Real Time Systems", International Edition, McGrawHill Companies, Inc., New York, 1997
2.	Jane W.S. Liu, "Real-Time Systems", Pearson Education India, 2000.
3.	Philip A. Laplante and Seppo J. Ovaska, "Real-Time Systems Design and Analysis: Tools for the Practitioner'' IV Edition IEEE Press, Wiley, 2013.

4.	Sanjoy Baruah, Marko Bertogna, Giorgio Buttazzo, "Multiprocessor Scheduling for Real-Time Systems ", Springer International Publishing, 2015

Course Outcomes (CO)

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

CO1	Gain knowledge about Schedulability analysis.
CO2	Learn about the Real-time programming environments.
CO3	Attain knowledge about real time communication and databases.
CO4	Develop a real time system.