MASTER OF SCIENCE (COMPUTER SCIENCE)

SYLLABUS
FOR
CREDIT-BASED CURRICULUM

(Applicable from 2022-2023 onwards)

Department of Computer Applications
National Institute of Technology
Tiruchirappalli

MASTER OF SCIENCE
(COMPUTER SCIENCE)

SYLLABUS

FOR

CREDIT-BASED FLEXIBLE CURRICULUM

(APPLICABLE FROM 2022-2023 ONWARDS)

DEPARTMENT OF COMPUTER
APPLICATIONS NATIONAL INSTITUTE OF
TECHNOLOGY TIRUCHIRAPPALLI-620 015,
TAMIL NADU - INDIA

M.Sc. SYLLABUS – CORE SUBJECTS

Semester	Subject Code	Subject Name	L	T	P	C
I	CAS 711	Mathematical Foundations of Computer Science	3	0	0	3
	CAS 713	Networking Technologies	2	1	0	3
	CAS 715	Data Structures and Algorithms	2	1	0	3
	CAS 717	Problem Solving using Python and R	2	1	0	3
	CAS 719	Operating Systems Fundamentals	3	0	2	4
	CAS 701	Data Structures Lab	0	0	4	2
	CAS 703	Python and R Lab	0	0	4	2
II	CAS712	Computer Organization and Architecture	3	0	0	3
	CAS 714	Theory of Computation	2	1	0	3
	CAS 716	Advanced Statistical Techniques for Data Science	3	0	0	3
	CAS 718	DBMS and Data Mining	3	0	0	3
	CAS 7AX	Elective I	3	0	0	3
	CAS 702	FOSS Lab	0	0	4	2
	CAS 704	DBMS and Data Mining Lab	0	0	4	2
III	CAS 721	Web Computing	2	1	0	3
	CAS 723	Artificial Intelligence and Machine Learning	3	0	0	3
	CAS 725	Object Oriented Software Engineering	3	0	0	3
	CAS 7BX	Elective II	3	0	0	3
	CAS 7CX	Elective III	3	0	0	3
	CAS 705	Artificial Intelligence and Machine Learning Lab Project Work Phase-I	0	0	4	2
	CAS 749	Project Work Phase-I	0	0	4	2
IV	CAS 750	Project Work Phase-II	-	-	-	10
		Grand Total	40	5	26	68

L: LECTURE | T: TUTORIAL | P: PRACTICAL | C: Credits

LIST OF ELECTIVES

Subject Code	Subject Name	L	T	P	C
ELECTIVE – I					
CAS 7A1	Internet of Things	3	0	0	3
CAS 7A2	Cyber Security	3	0	0	3
CAS 7A3	Cloud and Edge Computing	3	0	0	3
CAS 7A4	Mobile Computing	3	0	0	3
CAS 7A5	Cryptography and Blockchain Technology	3	0	0	3
CAS 7A6	Computer Graphics	3	0	0	3
ELECTIVE – II					
CAS 7B1	Big Data Analytics	3	0	0	3
CAS 7B2	Computational Intelligence	3	0	0	3
CAS 7B3	Computer Vision and Pattern Recognition	3	0	0	3
CAS 7B4	Soft Computing Techniques	3	0	0	3
CAS 7B5	Defensive and Secure Software Development	3	0	0	3
CAS 7B6	Software Verification and Validation	3	0	0	3
ELECTIVE – III					
CAS 7C1	GPGPU Programming	3	0	0	3
CAS 7C2	Design Patterns	3	0	0	3
CCS 7C3	Compiler Design	3	0	0	3
CAS 7C4	Mobile Application Development	3	0	0	3
CAS 7C5	Augmented Reality and Virtual Reality	3	0	0	3
CAS 7C6	Fault Tolerance Systems and Techniques	3	0	0	3

L: LECTURE | T: TUTORIAL | P: PRACTICAL | C: Credits

CAS 711 - MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

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

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

COURSE OUTCOMES:

Students will be able to

- Apply the concepts of discrete mathematics in the modelling and design of computational algorithms.

CAS 713 - NETWORKING TECHNOLOGIES

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

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

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”, 5th 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.
6. Chwan-Hwa Wu, and J. David Irwin, “Introduction to Computer Networks and Cybersecurity”, 1st Edition, CRC Press, 2013.

7. Black, Chuck., Goransson, Paul., and Culver, Timothy. “Software Defined Networks: A Comprehensive Approach”. Elsevier Science, 2016.

COURSE OUTCOMES:

Students will be able to:

- List the functionalities of networking layers available in both OSI reference model and TCP/IP model.
- Describe available LAN and WAN Technologies.
- Describe the principles of packet switching, forwarding, and routing.
- Distinguish between TCP and UDP packet formats.
- Describe the available application protocols and networking services.

CAS 715 - DATA STRUCTURES AND ALGORITHMS

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

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

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 quick sort – 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”, 4th Edition, Addison Wesely, 2011.
3. Steve S. Skiena, “The Algorithm Design Manual”, 2nd Edition, Springer, 2008.
4. George T. Heineman, Gary Policeand Stanley Selkow, “Algorithms in a Nutshell”, 2nd Edition, O’Reilley, 2016.
5. Kleinberg and Tardos, “Algorithm Design”, Pearson, 2006.
6. Anany Levitin, “The Design and Analysis of Algorithms”, 3rd Edition, Pearson, 2012.
7. Aditya Y. Bhargava, “Grokking Algorithms”, Manning Publications, 2016.

COURSE OUTCOMES:

Students will be able to

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

CAS 717 - PROBLEM SOLVING USING PYTHON AND R

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To introduce Python data structures, functions, and object-orientation concepts.
- To manipulate files using Python.
- To use Python packages for data science.
- To adopt R for statistical computing.

Python: variables, expressions, statements, operators, strings, Data structures: List, Dictionary, Tuples Functions: parameters, return values, local and global scope, recursion, lambda functions and function composition.

Object orientation – Classes, Objects, methods, Operator overloading, and Inheritance, Method resolution order. Files: Text files, reading and writing, Handling CSV, Json files, Object serialization, Errors, and Exception handling.

Python Modules and Packages: Python Standard Library- Libraries for Data science: Numpy, Pandas, Matplotlib, Scikit-Learn, SQLite.

R–Data sets - Data Visualization -Vector functions- Matrices and arrays- Matrix algebra-Random numbers- sampling and shuffling- Probability functions.

R for Statistical computing- Classical tests -Statistical modelling- Regression- Analysis of Variance- Analysis of Covariance- Bayesian Statistics.

REFERENCES:

1. John V. Guttag, “Introduction to Computation and Programming Using Python: with Application to Computational Modelling and Understanding Data”, 3rd Edition- MIT press- 2021.
2. Paul J. Deitel, and Harvey Deitel, “Python for Programmers” 1st Edition, Pearson, 2020.
3. Martin C. Brown, “Python: The Complete Reference”, Mc-Graw Hill, 2018.
4. Michael J. Crawley, “The R Book”, 2nd Edition, Wiley, 2013.
5. Thulin, Måns, “Modern Statistics with R: From Wrangling and Exploring Data to Inference and Predictive Modelling”, Eos Chasma Press, 2021.
6. Dalgaard, Peter, “Introductory Statistics with R”, Springer, 2008.

COURSE OUTCOMES:

Students will be able to:

- Develop solutions to real world problems using Python data structures, Functions and Object orientation concepts.
- Read and write data from/to files for processing using Python.
- Make use of Python Modules and Packages to solve complex problems.

- Perform statistical modelling using R.

CAS 719 - OPERATING SYSTEMS FUNDAMENTALS

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To understand design of an operating system and services provided by the OS.
- To understand what a process is and how processes are synchronized and scheduled.
- To acquire knowledge on different approaches to memory management.
- To understand the structure and organization of the file system and disk.
- To know the concepts of distributed and Mobile operating systems.

Operating System concepts - OS Structure – OS Services - System calls – Process management: Process Concept-Operations on process-Cooperating processes- Inter-process communication. Process scheduling-Scheduling 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. 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: UNIX, Linux, Android, and Windows Operating Systems.

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:

Students will be able to:

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

CAS 701 - DATA STRUCURES LAB

Exercises for learning basic features of C/C++ and exercises to implement various data structures for real world applications.

LAB OUTCOMES:

Students will be able to:

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

CAS 703 - PYTHON AND R LAB

Exercises for learning basic features of Python and R Programming.

LAB OUTCOMES:

Students will be able to:

- Write Python and R programs for problem solving.
- Use Python libraries for data science.
- Use R packages for statistical modelling.

CAS 712 - COMPUTER ORGANIZATION AND ARCHITECTURE

COURSE PRE-REQUISITES: Operating Systems Fundamentals (CAS 769)

COURSE OBJECTIVES:

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

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.
6. M. Morris Mano, "Digital design", Prentice Hall, 2002.

7. Hennessy, John L., Patterson, and David A., Asanović, Krste, “Computer Architecture: A Quantitative Approach”, Elsevier Science, 2019.

COURSE OUTCOMES:

Upon successful completion of the course, students will be able to:

- Understand basic structure and operation of digital computer.
- Understand the design of ALU to perform arithmetic and logic operations on fixed point and floating numbers.
- Understand different types of instructions and addressing modes supported in the instruction set of CPUs.
- Understand the design of control unit.
- Understand different types of memory devices, I/O communication techniques and standard I/O Interfaces.

CAS 714 – Theory of Computation

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

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

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's 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 GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, 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 accepters 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. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman (2007), "Introduction to Automata Theory Languages and Computation", 3rd edition, Pearson Education, India.

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:

Students will be able to

- Define languages by abstract, recursive definitions and regular expressions.
- Design a finite automaton to recognize a given regular language.
- Transform a language into regular expression or finite automaton or transition graph.
- Define deterministic finite automata and nondeterministic finite automata.
- Prove properties of regular languages and classify them.

CAS 716 - ADVANCED STATISTICAL TECHNIQUES FOR DATA SCIENCE

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To understand advanced statistical techniques.
- To gain comprehensive knowledge on applications of statistical techniques for data analysis.

Principles of statistical inference- Formulation of problems with examples- Point estimation. Estimator and estimate-Criteria for good estimates. Method of moments estimation and maximum likelihood estimation - Fisher Information matrix- Properties of maximum likelihood estimator- Confidence intervals.

Basic multivariate statistics: multivariate descriptive statistics, multivariate distributions (normal, etc), multivariate inferential statistic.

Multivariate data- Analysis of variance (ANOVA), Multivariate analysis of variance (MANOVA)- Case study: MANOVA

Multiple linear regression- Multiple and partial correlation- Detection of Collinearity-Stepwise regression.

Validation of model assumptions- Detection of outliers- influential observation and autocorrelation

REFERENCES:

1. Black, Bill., Hair, Joseph., Anderson, Rolph., and Babin, Barry. “Multivariate Data Analysis”. N.p.: Pearson Education, 2016.
2. R. A. Johnson and D. W. Wichern, “Applied multivariate statistical analysis”, Sixth Edition, PHI, 2012.
3. D. C. Montgomery and G. C. Runger, “Applied Statistics and Probability for Engineers”, 6th Edition, Wiley, 2014.
4. G.K. Bhattacharya and R.A. Johnson, “Statistical Concepts & Methods”, 6th Edition, Wiley, 2010.
5. Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining, “Introduction to Linear Regression Analysis”, 6th edition, Wiley, 2021.
6. W. J. Conover, “Practical Nonparametric Statistics”, 3rd Edition, Wiley, 1998.

COURSE OUTCOMES:

Students will be able to:

- Apply statistical techniques for real time data analysis applications

CAS 718 - DBMS AND DATA MINING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To learn different database models and design of databases
- To study query languages, transaction management, indexing and hashing
- To introduce the basic concepts and techniques of data mining
- To develop skills of using recent DBMS and data mining tools for solving problems.

Database system – Terminologies – Views – Data models – Database languages – Architecture – E-R Model – Extended E-R - Relational Model -Relational database design – Anomalies - Functional dependencies – 1NF to 5NF – Decomposition –Relational Query Languages – Relational Algebra – Tuple and domain Relational Calculus

SQL - Query processing and optimization – Transformation of relational expressions - Transaction – Properties – Concurrent execution – Serializability – Concurrency control – Protocols – Recovery System – Database Security - File organization.

Data Mining: Definition – Functionalities – Task Primitives – Integration with Database or Warehouse Systems – Issues – Data Sets – Attributes – Statistical Descriptions – Data Visualization – Measuring Data Similarity. Pre-processing: Data Reduction, Transformation, Discretization, Cleaning, and Integration.

Classification – Cluster Analysis – Outlier Analysis - Data Stream Mining, Mining Time Series, Text Mining, Data Stream Clustering, mining Big Data through data mining and analytical tools - Market Basket Analysis - Fuzzy Data Mining approaches – Rough Sets – Support Vector Machines – Genetic Algorithms.

Data warehousing – OLAP – Data Cubes - Mining Frequent Patterns, Associations and Correlations – Frequent Item set Mining Methods – Kinds of Association Rules – Constraint Based Frequent Pattern Mining.

REFERENCES:

1. Silberschatz, Abraham Korth, Henry F., and Sudarshan S., “Database System Concepts”, 7th Edition, McGraw Hill, 2021.
2. R.Elmasri, and S.B.Navathe, “Fundamentals of Database Systems”, 7th Edition, Pearson Education, 2017.
3. Guy Harrison, “Next Generation Databases”, Apress, 2015.
4. Adam Fowler, “NoSQL for dummies”, John Wiley & Sons, 2015.
5. David L. Olson and Dursun Delen, “Advanced Data Mining Techniques”, Springer, 2008.

6. Charu C. Aggarwal and Haixun Wang, “Managing and Mining Graph Data”, Springer, 2010.
7. Ian H. Witten, Eibe Frank and Mark A. Hall, “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann Publishers, 2011.
8. Jiawei Hanand, and Micheline Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, 2006.
9. Margaret H. Dunham, “Data Mining Introductory and Advanced Topics”, Prentice Hall, 2003.

COURSE OUTCOMES:

Students will be able to:

- Illustrate the features of DBMS and models for designing databases
- Apply logical database design principles in solving real world problems
- Understand the concepts, algorithms, and techniques of data mining

CAS 702 - FOSS LAB

Lab objective is to explore various open-source packages/technologies and apply them.

LAB OUTCOMES:

- To use GitHub for software development and version control in teams.
- To develop applications in FOSS environment.
- To identify and evaluate various FOSS options for any software requirement.

CAS 704 - DBMS AND DATA MINING LAB

- Exercises to construct and query databases.
- Exercises to implement data mining algorithms using ENCOG and WEKA.

LAB OUTCOMES:

Students will be able to:

- Work with ETL tools.
- Demonstrate classification, clustering in large data sets.
- Apply mining techniques for realistic data.

CAS 721 – WEB COMPUTING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

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

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 Servelet. 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”, 6th 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.
6. David Whiteley, “E-Commerce: Strategy, Technologies and Applications”, McGraw Hill Education, 2017.
7. Giulio Zambon, and Michael Sekler, “Beginning JSP, JSF and Tomcat Web Development”, Apress, 2007.

COURSE OUTCOMES:

Students will be able to:

- Design and develop web applications.
- Understand client-side and server-side scripting and their applicability.

CAS 723 - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To explore various AI search algorithms.
- To understand fundamentals of knowledge representation.
- To acquire knowledge on the basic concepts and techniques of Machine Learning.
- Introduce the facts and concepts of cognitive science by computational model and their applications.
- To gain knowledge on the applications of AI.

Introduction to Artificial Intelligence -Foundations of AI, Philosophy of artificial intelligence, problem solving, search techniques, constraint satisfaction, and game playing - minimax, Alpha beta pruning.

Knowledge Representation and Reasoning -Ontologies-foundations of knowledge representation and reasoning-representing and reasoning about objects-relations- events-actions- time- and space- predicate logic-situation calculus-description logics-reasoning with defaults, -reasoning about knowledge-sample applications.

Representing Knowledge and reasoning in an Uncertain Domain- Bayes rule-Bayesian networks-probabilistic inference-sample applications- Planning: planning as search- partial order planning- construction and use of planning graphs.

Machine learning- Supervised learning - Regression, Classification; unsupervised learning- Clustering; Reinforcement learning, Computational Intelligence- Fuzzy systems, Swarm intelligence, neural networks models- Learning through neural nets; Basics of Deep learning

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

REFERENCES:

1. Stuart Russell and Peter Norvig, “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”, 2nd Edition, CRC press, 2018.

COURSE OUTCOMES:

Students will be able to:

- Build simple knowledge-based systems.

- Apply knowledge representation and machine learning techniques to solve real-world problems.
- Apply computational Intelligence techniques.

CAS 725 - OBJECT ORIENTED SOFTWARE ENGINEERING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To comprehend basics of the software engineering process lifecycle.
- To introduce the object-oriented (OO) approach to software development through OO principles.
- To be conversant with UML (Unified Modelling Language) and the benefits of visual modelling/ diagramming.
- To explore software engineering principles for both procedural and object-oriented approaches.

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

Software Requirements Elicitation and Analysis - Software Requirement – Requirements Elicitation Techniques – Characteristics - Software Requirements Specification (SRS) Document - Requirements Change Management - Object Oriented Analysis - Overview of Cost Estimation Techniques - Agile development – Classification of methods – Agile project management - Lifecycle – Work products, Roles, and Practices values – Process mixtures– Adoption strategies – Understanding SCRUM.

Software Design - Object Oriented Design - UML-Refinement of Use Case Description – Refinement of classes and relationships - Construction of class diagrams - Development of Details – Design and Creation–Generating Test cases from User Cases – Object Oriented Design principles.

Software Implementation - Quality and Metrics - Software Implementation – Tools and Techniques – Software quality – Models - Measurement basic - analyzing the metric data-Metrics for measuring size and structure – Measuring software quality - Object oriented metrics – Overview of Scala.

Software Testing and Maintenance - Software verification techniques – Checklist: a popular – verification tool -Functional Testing – Structural Testing – Object Oriented Testing - Class testing–State based testing – Mutation testing - Levels of testing -Tools - Software maintenance – Categories – Challenges – Maintenance of Object-oriented Software – Software rejuvenation – Configuration management – Regression testing.

REFERENCES:

1. Roger Pressman, and Bruce R. Maxim, “Software Engineering: A Practitioner's Approach”, McGraw-Hill Higher Education, 9th Edition, 2020.
2. Yogesh Singh, Ruchika Malhotra, “Object-Oriented Software Engineering”, PHI,2012.
3. Timothy C. Lethbridge and Robert Laganiere, “Object-Oriented Software Engineering”, McGraw-Hill, 2nd ed., 2004.

4. Grady Booch, “Object-Oriented Analysis and Design with Applications”, 3rd Edition, Pearson Education, 2009.
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.

COURSE OUTCOMES:

Students will be able to:

- Practice the principles of object-oriented software development and various CASE tools.
- Convey design decisions using UML.
- Compute cost of software development and measure quality of software.

CAS 705 - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB

Exercises to implement

- Various AI and ML Techniques.

LAB OUTCOME:

Students will be able to:

- Apply the various search techniques.
- Apply knowledge representation and reasoning techniques.

CAS749 PROJECT WORK –Phase I

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

Outcome: To publish papers in conference or journal.

CAS 750 PROJECT WORK –Phase II

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

Outcome: To publish papers in conference or journal.

CAS 7A1 – INTERNET OF THINGS

COURSE OBJECTIVES:

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

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

COURSE OUTCOMES:

Students will be able to:

- To understand the working of IoT devices.
- To program the IoT devices.
- To develop applications using IoT devices.

CAS 7A2 - CYBER SECURITY

COURSE OBJECTIVES:

- To study the concepts and principles of Cyber Security.
- To understand the security threats and the ways to mitigate them.
- To study the effectiveness of network and data security toolkits.

Terminology of security - CIA - Security Models - Security and Privacy, Secure software development – Attack and Mitigation – Vulnerability, Threat, Risk

Cryptography: Classical Cryptography, Modern Cryptography – Forensics -watermarking - finger printing – Steganography - Biometrics

Network Security - Wireless Security, IDS and IPS, Network Intrusion Management Tools, Unified Threat management (UTM), Malware

Application Security, Web application firewall (WAF), Software Security, Mobile Security, Edge Security.

Data Security, Block chain, Bitcoins, Hacking Tools, and Techniques.

REFERENCES:

1. W. Stallings, "Cryptography and Network Security: Principles and Practice", 7th Edition, Prentice Hall, 2017
2. Kutub Thakur, and Al-Sakib Khan, "Cybersecurity Fundamentals, A Real-World Perspective", ISBN 9780367472504, CRC Press, 2020.
3. Raef Meeuwisse, "Cybersecurity for Beginners", Cyber Simplicity Ltd; 2nd edition, 2017.
4. Gautam Kumar, Dinesh Kumar Saini, and Nguyen Ha HuyCuong, "Cyber Defense Mechanisms, Security, Privacy, and Challenges", CRC Press - ISBN 9780367408831, 2020.

COURSE OUTCOMES:

Students will be able to:

- To understand fundamentals of cyber security.
- To study and analyze cryptographic and forensic methods.
- Analyze the network and application security issues.
- Explore the nature and logic behind security threats on the cyber space as an ethical hacker.

CAS 7A3 - CLOUD AND EDGE COMPUTING

COURSE OBJECTIVES:

- To understand Cloud Computing concepts, technologies, architecture, and applications.
- To understand different cloud programming platforms and tools to develop and deploy applications on cloud.
- To understand edge computing architecture and technologies.

Overview of Computing Paradigms- Distributed Systems Models and Enabling Technologies - Cloud Computing Properties and Characteristics - Business Drivers for Adopting Cloud Computing

Introduction to virtualization - Different Approaches to Virtualization - Server, Storage, Network Virtualization - Virtual Machine Provisioning and Manageability - VM Placement- VM Migration - Hypervisors - Case studies: VMware, KVM, Xen – Containers

Service Science - Service Oriented Architecture - Web Services: SOAP, WSDL, UDDI - Web Services Discovery and Composition - REST based Web Services

Cloud Computing Architecture - Cloud Computing Service Delivery Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) Deployment Models: Public cloud, Private cloud, Hybrid cloud - Data Centre Design and Management - Service Level Agreements (SLAs) – Pricing Models of Cloud - Migrating to Cloud – Cloud Simulators - Cloud Security Risks - Case Studies: Amazon AWS, Microsoft Azure, Amazon EC2, Google Cloud

Edge computing Concepts - Reference Architecture - Edge Computing in the IoT–Edge Analytics and Data Management -Use Cases

REFERENCES:

1. Kai Hwang, Jack Dongarra, and Geoffrey C. Fox, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things”, Morgan Kaufmann, 2013.
2. Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju, “Web Services: Concepts, Architectures and Applications”, Springer, 2010.
3. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill, 2017.
4. Ajit Singh, “Edge Computing”, Shroff Publishers, 2019.
5. Perry Lea, “IoT and Edge Computing for Architects”, Packt publishers, 2020.

COURSE OUTCOMES:

Students will be able to:

- Acquire Knowledge on the concepts and technologies of Cloud Computing.

- Define the principles of virtualization
- Identify the Service Oriented Architecture for Distributed Computing workflow.
- Acquire Knowledge on the concepts and technologies of Edge Computing.

CAS 7A4 – MOBILE COMPUTING

COURSE OBJECTIVES:

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

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 Telukder, 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 by Pearson, 2016.
4. Kumkum Garg, “Mobile Computing Theory and Practice”, Pearson, 2010.
5. Behrouz A Forouzan, “TCP/IP Protocol Suite”, 3rd Edition by TMH, 2008.

COURSE OUTCOMES:

Students will be able to:

- Understand concepts of Mobile Communication.
- Analyse next generation Mobile Communication System.
- Understand network and transport layers of Mobile Communication.
- Analyze various protocols of all layers for mobile and ad hoc wireless communication networks.
- Understand IP and TCP layers of Mobile Communication.

CAS 7A5 - CRYPTOGRAPHY AND BLOCKCHAIN TECHNOLOGY

COURSE OBJECTIVES:

- To understand the basic network security concepts.
- To acquire knowledge of cryptographic algorithms.
- To understand the history, types, and applications of Blockchain.
- To acquire knowledge about cryptography and consensus algorithms.

Introduction - Security Trends-The OSI Security Architecture, Security Attacks, Security Services and Security Mechanisms, A model for Network Security-Classical Encryption Techniques.

Cryptographic algorithms - Number Theory- Modern Block Ciphers: DES, 3DES, AES, Blowfish, CAST-128 - Stream Cipher - Public Key Cryptography: RSA, Diffie-Hellman, Elgamal, ECC.

Data Integrity Algorithms - MD5 message digest algorithm - Secure hash algorithm (SHA) Digital Signatures: Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security - pretty good privacy (PGP) - S/MIME.

Introduction to Blockchain - Distributed DBMS – Limitations of Distributed DBMS, Introduction to Block chain – History, Definition, Distributed Ledger, Blockchain Categories – Public, Private, Consortium, Blockchain Network and Nodes, Peer-to-Peer Network, Mining Mechanism, Generic elements of Blockchain, Features of Blockchain, and Types of Blockchain.

Blockchain Architecture - Operation of Bitcoin Blockchain, Blockchain Architecture – Block, Hash, Distributer P2P, Structure of Blockchain- Consensus mechanism: Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), Proof of Authority (PoA) and Proof of Elapsed Time (PoET)

REFERENCES:

1. William Stallings, “Cryptography and Network Security Principles and Practices”, Pearson/PHI, 2017.
2. William Stallings, “Network Security Essentials (Applications and Standards)”, Pearson Education, India, 2017.
3. W. Mao, “Modern Cryptography – Theory and Practice”, Pearson Education, 2011.
4. Charles P. Pfleeger, and Shari Lawrence Pfleeger, “Security in computing”, Prentice Hall of India, 2015.
5. Imran Bashir, “Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained”, 2nd Edition, Packt Publishing Ltd, March 2018.

6. Bellaj Badr, Richard Horrocks, and Xun (Brian) Wu, “Blockchain by Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger”, Packt Publishing Limited, 2018.

COURSE OUTCOMES:

Students will be able to:

- Recall the basic network security concepts.
- Apply Cryptographic Algorithms
- Describe the history, types, and applications of Blockchain.
- Gain familiarity with cryptography and Consensus algorithms.

CAS 7A6 - COMPUTER GRAPHICS

COURSE OBJECTIVES:

- To provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.
- To understand computer graphics techniques (2-D/3-D), focusing on 3D modelling, image synthesis, and rendering.
- Introduce geometric transformations, geometric algorithms, software systems (OpenGL), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis, shading and mapping, ray tracing, radiosity, global illumination, photon mapping, and anti-aliasing.
- To explore the interdisciplinary nature of computer graphics which is emphasized in the wide variety of examples and applications.

Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Midpoint circle generating algorithm, and parallel version of these algorithms.

Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections, and shearing. Windowing and Clipping: Viewing pipeline, viewing transformations, 2-D Clipping Algorithms-Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against nonrectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping, Text clipping.

Compression - Text and Image Compression: Introduction, compression principles, text compression, image compression, Audio and Video Compression: Introduction, audio compression, DPCM, ADPCM, APC, LPC, video compression, video compression principles, H.261, H.263, MPEG, MPEG-1, MPEG-2, and MPEG-4

3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping, Quadric surfaces, Spheres, Ellipsoid, Blobby objects, introductory concepts of Spline, Bspline and Bezier curves and surfaces.

Back Face Detection algorithm, Depth buffer method, A-buffer method, Scan line method, basic illumination models – Ambient light, Diffuse reflection, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Color consideration, Transparency and Shadows.

REFERENCES:

1. Hearn, Donald., and Baker, M. Pauline, “Computer Graphics, C Version”. India: Pearson Education, 1997.
2. Motta, G., Bryant, D., and Salomon, David, “Data compression: the complete reference”, 4th Edition, Springer- Verlag London, 2007.

3. Ralf Steinmetz, and Klara Narstedt , "Multimedia Fundamentals: Vol 1 - Media Coding and Content Processing", Pearson Education, 2004.
4. Singh, Amarendra N.., Udai, and Arun D., "Computer Graphics", Tata Mcgraw Hill Education Private Limited, 2008.
5. Harrington, Steven. "Computer Graphics: A Programming Approach" McGraw-Hill, 1987.
6. Davd F. Rogers, "Procedural Elements for Computer Graphics", McGraw-Hill Education (India) Pvt Limited, 2001.

COURSE OUTCOMES:

Students will be able to:

- Understand concepts of computer graphics.
- Understand different computer graphic techniques.

CAS 7B1 - BIG DATA ANALYTICS

COURSE OBJECTIVES:

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

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. DT Editorial Services, "Big Data Black Book: (Covers Hadoop 2, MapReduce, Hive, YARN, Pig, R and Data Visualization)", India: Dreamtech Press, 2016.
2. Tom White, "Hadoop: The Definitive Guide", O'reilly Media, 4th Edition, 2015.
3. 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.
4. Jure Leskove, Anand Rajaraman, and Jeffrey D. Ullman, "Mining of Massive Datasets", Cambridge University Press, 3rd Edition, 2020.
5. Guy Harrison, "Next Generation Databases: NoSQL and Big Data", APress, 1st Edition, 2015.
6. 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.

COURSE OUTCOMES:

Students will be able to:

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

CAS 7B2 - COMPUTATIONAL INTELLIGENCE

COURSE OBJECTIVES:

- To introduce the fundamentals of key intelligent technologies including neural networks, fuzzy systems, evolutionary computation, and swarm intelligence.
- To explain the integration of intelligent systems.

Introduction to Computational Intelligence - Intelligence machines - Computational Intelligence paradigms – Fuzzy logic - Fuzzy relationships - Fuzzy Sets - Operations on Fuzzy sets - Fuzzy rules - Fuzzy inference systems - Fuzzy expert systems - Applications of Fuzzy Set theory to different branches of science and engineering

Neural Network - Biological foundation of Neural Network - Neural Model - Network Architectures - Perceptron Learning - Supervised and unsupervised learning neural networks - Hebbian Learning - Back-propagation - Associative Learning - Competitive Networks - Hopfield Network - Deep neural networks and learning algorithms - Applications - Case studies

Neuro Fuzzy Systems - Adaptive Neuro - Fuzzy Inference Systems - Architecture - Hybrid Learning Algorithm - Learning Methods that Cross-fertilize ANFIS and RBFN - Coactive Neuro Fuzzy Modelling - Framework Neuron Functions for Adaptive Networks - Neuro Fuzzy Spectrum

Evolutionary computation – Chromosomes - Fitness functions - Selection mechanisms - Genetic algorithms - Crossover - mutation – Convergence – Applications - Genetic programming - Evolution strategies - Evolutionary neural network - Case studies

Swarm Intelligence - Foundations - Examples – Metaheuristics - ACO method - Ant System – Birds – PSO – Firefly Algorithm - Applications - Case Studies

REFERENCES:

1. Shi, Yuhui., and Eberhart, Russell C., “Computational Intelligence: Concepts to Implementations”, Elsevier Science, 2011.
2. S. Haykin, “Neural Networks – A Comprehensive Foundation”, Prentice Hall, 1999.
3. N. Sivanandam, and S. N. Deepa, “Principals of soft Computing”, Wiley India, 2nd ed., 2011.
4. A.P. Engelbrecht, “Computational Intelligence: an Introduction”, 2nd Edition, John Wiley & Sons, 2012.
5. H.K. Lam, S.S.H. Ling, and H.T. Nguyen, “Computational Intelligence and Its Applications: Evolutionary Computation, Fuzzy Logic, Neural Network and Support Vector Machine”, Imperial College Press, 2011.
6. J.S.R. Jang, C.T. Sun and E. Mizutani, “Neuro-Fuzzy and Soft Computing”, PHI, 2004.
7. J. Freeman and D. Skapura, “Neural Networks: Algorithms, Applications, and Programming Techniques”, Addison-Wesley, 1991.

8. G. J. Klir, and B. Yuan, “Fuzzy Sets and Fuzzy Logic: Theory and Applications”, Prentice Hall, 1995.
9. S. Rajasekaran and G.A.V. Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms”, PHI, 2003.

COURSE OUTCOMES:

Students will be able to:

- Apply intelligent technologies for various applications.
- Model global optimization solutions for various real-life problems.

CAS 7B3 - COMPUTER VISION AND PATTERN RECOGNITION

COURSE OBJECTIVES:

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

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

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, Mean-Shift, 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”, 2nd 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”, 3rd Edition, Prentice Hall of India, 2008.
5. K.Pratt, “Digital Image Processing”, 4th Edition, McGraw Hill, 2007.
6. R. O. Duda, P. E. Hart and D. Stork, “Pattern Classification” 2nd Edition, Wiley, 2002.
7. C. Bishop, “Pattern Recognition and Machine Learning”, Springer 2006.

COURSE OUTCOMES:

Students will be able to:

- Apply fundamental algorithms in Image Processing and analyze their applicability for real time problems.
- Design solutions for various computer vision and pattern recognition problems.

CAS 7B4 - SOFT COMPUTING

COURSE OBJECTIVES:

- To learn the techniques of soft computing.
- To learn the applications of evolutionary and genetic algorithms.
- To know the design of fuzzy controller and rough sets.
- To know the hybridization of soft computing systems.

Soft Computing: Introduction, Soft Computing verses Hard Computing, Need, Tools and techniques, Applications

Artificial Neural Network (ANN): 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 Artificial Neural Network

Evolutionary and Genetic Algorithms: Concepts, Operators, Function Optimization, Dominance, Swarm Intelligence, Colony systems, Modeling Collective Behaviour in Social Insects, Division of Labor and Task Allocation, Applications.

Fuzzy Logic: Fuzzy Sets, Operations on Fuzzy sets, Fuzzy Relations, Fuzzy Systems, Fuzzy controller, Fuzzy Decision Making, Fuzzy Clustering, Applications. Rough sets: Imprecise Categories Approximations and Rough Sets, Reduction of Knowledge, Decision Tables, and Applications.

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”, PHI, 2015.
2. G. J. Klir, and B. Yuan, “Fuzzy Sets and Fuzzy Logic: Theory and Applications”, 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”, Pearson Education, 1989.
5. E. Bonabeau, M. Dorigo, and G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems”, Oxford Press, 1999.
6. K. L. Du, and M. N. S. Swamy, “Neural Networks in a Soft computing Framework”, Springer, 2008.
7. R. A. Aliev, and R. R. Aliev, “Soft Computing and Its Applications”, World Scientific, 2001.

8. D. Ruan, “Intelligent Hybrid Systems”, Kluwer Academic Publisher, 1997.

COURSE OUTCOMES:

Students will be able to:

- Explain the basics of soft computing.
- Apply neural network principles and algorithms for given problems.
- Apply the principles of fuzzy and hybrid algorithms for real time applications.
- Solve problems using evolutionary algorithms.

CAS 7B5 - DEFENSIVE AND SECURE SOFTWARE DEVELOPMENT

COURSE OBJECTIVES:

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

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.
6. LeBlanc, David., and Howard, Michael, “Writing Secure Code”, Microsoft Press, 2002.

COURSE OUTCOMES:

Students will be able to:

- Understand software security fundamentals.
- Develop secure software.
- Use secure coding practices in software development.

CAS 7B6 - SOFTWARE VERIFICATION AND VALIDATION

COURSE OBJECTIVES:

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

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: Graph-Based 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 for Testing, Metrics for Maintenance, Software rejuvenation, Reusability and Portability, Estimation of maintenance efforts, Configuration management, Regression testing, Emerging Technologies.

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, 2 nd 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:

Students will be able to:

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

CAS 7C1 - GPGPU PROGRAMMING

COURSE OBJECTIVES:

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

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, Drop-In Libraries, OpenACC Directives

Parallel Programming Paradigms: Overview, Element Addressing - Multidimensional Kernel, Map, Gather, 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, Pinned.

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. Shane Cook, “CUDA Programming - A Developer’s Guide to Parallel Computing with GPUs”, Morgan Kaufmann Publishers, 2012.
4. Jason Sanders and Edward Kandrot, “CUDA by Example”, Addison Wesley, 2010

COURSE OUTCOMES:

Students will be able to:

- Analyze GPU Design for finding solutions.
- Understand parallel programming paradigms.
- Solving high performance computing problems using GPUs.

CAS 7C2 – DESIGN PATTERNS

COURSE OBJECTIVES:

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

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:

Students will be able to:

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

CAS 7C3 - COMPILER DESIGN

COURSE OBJECTIVES:

- To learn various phases of compiler.
- To learn various parsing techniques.
- To understand intermediate code generation techniques and run-time environment.
- To learn various code optimization techniques

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 precedenceparser – 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 data-flow 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:

Students will be able to:

- Explain various phases of compiler.
- Explain various parsing techniques.
- Explain the process of intermediate code generation.
- Explain various code optimization techniques.
- Apply the knowledge of lex tool and yacc tool to develop a scanner and parser, respectively.

CAS 7C4 – MOBILE APPLICATION DEVELOPMENT

COURSE OBJECTIVES:

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

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.

COURSE OUTCOMES:

Students will be able to:

- Describe the requirements for mobile applications.
- Develop design for mobile applications for specific requirements.
- Implement the design using mobile application development frameworks.

CAS 7C5 - AUGMENTED REALITY AND VIRTUAL REALITY

COURSE OBJECTIVES:

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

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:

Students will be able to:

- Understand the basic concepts of Augmented Reality and Virtual Reality (AR and VR).

CAS 7C6 FAULT TOLERANCE SYSTEMS AND TECHNIQUES

COURSE OBJECTIVES:

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

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 Check pointing; 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”, 2nd 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:

Students will be able to:

- Understand the basic concepts of fault tolerance.
- Practice the application principles of fault tolerance.
- Analyze the fault tolerance in hardware and software level.