GOVERNMENT AUTONOMOUS COLLEGE, ANGUL
MASTER OF SCIENCE DEGREE COURSE

M.Sc. COMPUTER SCIENCE

The Course of Study and the Scheme of Examinations

Year/ Semester	Subject	Paper	Title of Paper	Credit	Max. Marks		
					IA	Uni. Exam	Total
I Year I Semester	Core	CS-1.1	Data Structure and Algorithms	4	30	70	100
	Core	CS-1.2	Computer System Architecture	4	30	70	100
	Core	CS-1.3	Database Systems & Implementation	4	30	70	100
	Core	CS-1.4	Discrete Mathematical Structures	4	30	70	100
	Elective I	CS-1.5	(a) Data Analysis using Python (or) (b) Object Oriented Design using UML	4	30	70	100
	Core Practical	CS-1.6	Algorithms Lab	4		100	100
	Core Practical	CS-1.7	Database Lab	4		100	100
I Year II Semester	Core	CS-2.1	Computer Networks	4	30	70	100
	Core	CS-2.2	Advanced JAVA	4	30	70	100
	Core	CS-2.3	Operating System Design	4	30	70	100
	Core	CS-2.4	Theory of Computation	4	30	70	100
	Elective II	CS-2.5	(a)Data Mining (or) (b)Computer Graphics	4	30	70	100
	Core Practical	CS-2.6	JAVA Programming Lab	4		100	100
	Core Practical	CS-2.7	Operating Systems Lab	4		100	100
II Year III Semester	Core	CS-3.1	Artificial Intelligence	4	30	70	100
	Core	CS-3.2	Software Engineering	4	30	70	100
	Core	CS-3.3	Compiler Design	4	30	70	100
	Elective III	CS-3.4	(a) Information Security (or) (b) Cloud Computing	4	30	70	100
	Elective IV	CS-3.5	(a) Internet of Things (or) (b) Machine Learning	4	30	70	100
	Core Practical	CS-3.6	AI Programming Lab	4		100	100
	Core Practical	CS-3.7	Software Engineering Lab	4		100	100
II Year IV Semester		CS-4.1	Comprehensive Viva	4		100	100
		CS-4.2	Project Work and Viva Voce	12		300	300
			Total	**100**			**2500**

CS.1.1 DATA STRUCTURE & ALGORITHMS

OBJECTIVES

- To understand the fundamentals of different data structure.
- To be able to learn design principles and concepts of algorithms.
- To have a mathematical foundation in analysis of algorithm.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Learn the basic types for data structure, implementation and application.
- Know the strength and weakness of different data structures.
- Use the appropriate data structure in context of solution of given problem.
- Develop programming skills which require solving given problem.

UNIT-I

Introduction, The Role of algorithms in computing, Growth of functions, Recurrences, Heapsort, Quicksort, Sorting in linear time.

UNIT-II

Elementary Data structures, Hash Tables, Binary Search Trees, Red-Black trees, B-trees, Data Structures for Disjoint sets.

UNIT-III

Elementary Graph algorithms, Representation of Graphs, BFS, DFS, And Topological Sort, Minimum Spanning Trees Shortest path (single source and all-Pairs), Maximum Flow.

UNIT- IV

Dynamic programming (Matrix Chain, TSP Optimal binary Search) Greedy algorithms, Amortized analysis, String Matching.

UNIT- V

P, NP and NP-Completeness, Approximate Algorithm, Computational Geometry.

Text books:

T.H.Corman, C.E.Leiserson, R.L.Rivest and C. Stein : Introduction to Algorithms

Reference books:

1. Gilles Brassard and Paul Bratley: Fundamentals of Algorithmics
2. A.V. Aho, J.E.Hopcroft and J.D.Ullman: The Design and Analysis of Computer Algorithms

CS.1.2 COMPUTER SYSTEM ARCHITECTURE

OBJECTIVES

- To understand the structure, function and characteristics of computer systems.
- To understand the design of the various functional units and components of computers.
- To identify the elements of modern instructions sets and their impact on processor design.
- To explain the function of each element of a memory hierarchy in order to identify and compare different methods for computer I/O.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- The student will be able to understand the major architectural styles and appreciate the compromises that they encapsulate.
- They will be able to read outline descriptions of real processors and understand in which way their designs fit into the frameworks described in the course.
- They will be also able to understand the impact of design choices in programming in the context of a specific architecture.

UNIT-I

Computer Function and Interconnection: Computer Components, Computer Function, Interconnection Structures, Bus Interconnection, PCI. Cache Memory: Computer Memory System, Cache Memory Principles, Elements of Cache Design, Pentium-4 Cache Organization.

UNIT-II

External Memory : Magnetic Disk, RAID, Optical Memory, Magnetic Tape, External Devices, I/O Module, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels and Processors, FireWire and InfiniBand.

UNIT-III

CPU Structure and Function: Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining, The Pentium Processor. Reduced Instruction Set Computer (RISC): Instruction Execution Characteristics, Use of a Large Register File, Compiler- Based Register Optimization, Reduced Instruction Set Architecture, RISC Pipelining, MIPS R4000, SPARC, RISC versus CISC Controversy.

UNIT-IV

Instruction-Level Parallelism and Superscalar Processors: Overview, Design Issues, Pentium-4. IA-64 Architecture: Motivation, General Organization, Prediction, Speculation, and Software Pipelining, IA-64 Instruction Set Architecture, Itanium Organization.

UNIT-V

Parallel Organization: Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and MESI Protocol, Clusters, Non-Uniform Memory Access (NUMA), Vector Computation.

Text Book:

1. Stallings, W. Computer Organization and Architecture 4/ed. (PHI)

Reference Books

1. Mano. M . M.: Computer System Architecture 3/ed. (PHI)
2. Hayes, J.P.: Computer Architecture and Organization 3/ed. (Mc. Graw-Hill Int.)
3. Quinn, M. J.: Parallel Programming in C with MPI and OpenMP (TMH)

CS.1.3 DATABASE SYSTEMS & IMPLEMENTATION

OBJECTIVES

- To learn the fundamental elements of database system.
- To learn the basic concepts of relational database management systems.
- To learn various SQL commands.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify advance database concepts and database models.
- Apply and analyze various terms related to transaction management in centralized and distributed database.
- Produce data modeling and database development process for object–oriented DBMS.
- Analyze and Implement the concept of object- relational database in development of various real time software.

UNIT-I

Database System : Database System Applications, Database Systems versus File Systems, View of Data & Data Models, Database Languages, Database Users and Administrators, Transaction Management, Database System Structure, Application Architecture. Entity-Relationship Model : Basic Concepts & Constraints, Keys, Design Issues, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R Features, Design of E-R Database Schema, Reduction of an E-R Schema to Tables, Overview of Relational Model and Relational Database Design.

UNIT-II

SQL : Basic Structure, Set Operations, Aggregate Functions, Null Values, Nested Sub-queries, Views, Complex Queries, Modification of the Database, Joined Relations, Data-Definition Language, Embedded SQL. Dynamic SQL. Integrity and Security: Domain Constraints, Referential Integrity, Assertions, Triggers, Security and Authorization, Authorization in SQL, Encryption and Authentication.

UNIT-III

Query Processing: Measures of Query Cost, Selection Operation, Sorting, Join and other Operations, Evaluation of Expressions. Query Optimization: Estimating Statistics of Expression Results, Transformation of Relational Expressions, Choice of Evaluation Plans, Materialized Views.

UNIT-IV

Object-Oriented Databases: Complex Data Types, Object-Oriented Data Model, Object-Oriented Languages, Persistent Programming Languages, Persistent C++ Systems, Persistent Java Systems. Object-Relational Databases: Nested Relations, Complex Types, Inheritance, Reference Types, Querying with Complex Types, Functions and Procedures, Object-Oriented Vs Object-Relational.

UNIT-V

Transactions: Transaction, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Transaction Definition in SQL, Testing for Serializability. Concurrency Control: Lock-Based, Timestamp-Based, Validation-Based Protocols Multiple Granularity, Multiversion Schemes, Deadlock Handling, Insert and Delete Operations, Weak Levels of Consistency, Concurrency in Index Structures. Recovery System : Failure Classification, Storage Structure, Recovery and Atomicity, Log-Based Recovery, Shadow Paging, Recovery with Concurrent Transactions, Buffer Management, Failure with Loss of Nonvolatile Storage, Advanced Recovery Techniques, Remote Backup Systems.

Text Book: Silbcrschatz.A, Korth,H.F., Sudarshan.S. : Database System Concepts 4/ed (TMH)

CS.1.4 DISCRETE MATHEMATICAL STRUCTURES

OBJECTIVES

- To learn the mathematical foundations for Computer Science.
- Topics covered essential for understanding various courses.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Express a logic sentence in terms of predicates, quantifiers, and logical connectives.
- Apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.
- Use tree and graph algorithms to solve problems.
- Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.

UNIT-I

Fundamentals of logic, Prepositional equivalences, Predicates and Quantifiers, Nested Quantifiers, Methods of Proof, Sequences and summations, Mathematical Induction.

UNIT-II

Sets, set operations, properties of binary relations, equivalence relations and partitions, partial ordering relations and lattices, chains and anti-chains, functions and the pigeonhole principle.

UNIT-III

The basics of counting, permutations and combinations, recurrence relations, solving recurrence relations, generating functions, inclusion – exclusion

UNIT-IV

Introduction to graphs, graph terminology, Representing graphs and graph isomorphism, Euler and Hamilton paths, introduction to trees, applications of trees.

UNIT-V

Groups, subgroups, cosets and Lagrange's Theorem, Codes and group codes, homomorphisms and normal subgroups, Isomorphisms, Ring, Integral Domains and Fields.

Text Book:

1. C.L. Liu, "Elements of Discrete Mathematics", Mc Graw Hills International Second Edition.
2. Kenneth H. Rosen, Discrete Mathematics and its Applications, Mc Graw Hills International Fifth Edition.

Reference Books:

1. Bernardi Kolman, Robert C. Busby, Sharon Ross, "Discrete Mathematical Structure" Prentice Hall of India.
2. Mott, J.L, Kandel, A. & Baker, T.P..: Discrete Mathematics for Computer Science and Mathematics, 2/ed (P 1999)
3. N.Ch. S.N. Lyengar, Chankrasekaran, Venkatesh, Arunachalam, "Discrete Mathematics", Vikas Publication.

CS.1.5 (a) DATA ANALYSIS USING PYTHON

OBJECTIVES

This course is designed to teach students how to analyze different types of data using Python. Students will learn how to prepare data for analysis, perform simple statistical analysis, create meaningful data visualizations and predict future trends from data.

LEARNING OUTCOMES

- Understanding basics of python for performing data analysis
- Understanding the data, performing preprocessing, processing and data visualization to get insights from data.
- Use different python packages for mathematical, scientific applications and for web data analysis.
- Develop the model for data analysis and evaluate the model performance.

UNIT-I

Python Fundamentals for Data Analysis: Python data structures, Control statements, Functions, Object Oriented programming concepts using classes, objects and methods, Exception handling, Implementation of user-defined Modules and Package, File handling in python.

UNIT-II

Introduction to Data Understanding and Pre-processing: Knowledge domains of Data Analysis, understanding structured and unstructured data, Data Analysis process, Dataset generation, Importing Dataset: Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values.

UNIT-III

Data Processing and Visualization: Data Formatting, Exploratory Data Analysis, Filtering and hierarchical indexing using Pandas. Data Visualization: Basic Visualization Tools, Specialized Visualization Tools, Seaborn Creating and Plotting Maps.

UNIT-IV

Mathematical and Scientific applications for Data Analysis: Numpy and Scipy Package, Understanding and creating N-dimensional arrays, Basic indexing and slicing, Boolean indexing, Fancy indexing, Universal functions, Data processing using arrays, File input and output with arrays.

UNIT-V

Analysing Web Data: Data wrangling, Web scrapping, Combing and merging data sets, Reshaping and pivoting, Data transformation, String Manipulation, case study for web scrapping.

Text books:

1. Reema Thareja, “Python Programming using Problem Solving approach”, Oxford University press

Reference books:

1. David Ascher and Mark Lutz, Learning Python, Publisher O’Reilly Media.
2. Wes Mckinney “Python for Data Analysis”, First edition, Publisher O’Reilly Media.

CS.2.1 COMPUTER NETWORKS

OBJECTIVES

- To learn about computer network organization and implementation.
- To obtain a theoretical understanding of data communication and computer networks.
- To gain practical experience in installation, monitoring, and troubleshooting of current LAN systems

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Describe how computer networks are organized with the concept of layered approach.
- Describe how signals are used to transfer data between nodes.
- Implement a simple LAN with hubs, bridges and switches.
- Describe how packets in the Internet are delivered.

UNIT-I

Encoding & Modulation: Digital-To-Digital, Analog-to-Digital, Digital-to-Analog and Analog-t-Analog Conversions. Transmission of Digital Data, Interfaces and Modems: Digital Data Transmission, DTE-DCE Interface Standards, Modems, 56K Modem, Cable Modem. Multiplexing: Frequency Division, Wave Division and Time Division Multiplexing, Multiplexing in the Telephone System, Digital Subscriber Line (DSL), FTTC.

UNIT-II

Data Link Control: Line Discipline, Flow Control, Error Control. Data Link Protocols: Asynchronous Protocols, Character-Oriented Protocols, Bit-Oriented Protocols, Link Access Procedures. Local Area Networks: Project 802, Ethernet, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, Token Bus, Token Ring, Fiber Distributed Data Interface (FDDI).

UNIT-III

Metropolitan Area Networks: IEEE 802.6 (DQDB), Switched, Multimegabit Data Services (SMDS). Switching: Circuit Switching, Packet switching, Message Switching. Point-To-Point Protocol: Transition States, PPP Layers, Link Control Protocol (LCP), Authentication, Network Control Protocol (NCP).

UNIT-IV

Integrated Services Digital Network: Services, Scribers Access to the ISDN, ISDN layers, Broad Band ISDN. X.25 : X.25 Layers. Frame Relay: Frame Relay Operation, Frame Relay Layers, Congestion Control, Leaky Bucket Algorithm, Traffic Control.

UNIT-V

ATM: ATM Architecture, Switching, Switch Fabrics, ATM Layers, Service Classes, ATM Applications. SONET: Synchronous Transport Signals, Physical Configuration, SONET Layers, SONET Frame, Multiplexing STS Frames, Applications. Networking & Internetworking Devices: Repeaters, Bridges, Routers, Gateways, Routing Algorithms (Distance Vector & Link State Routing).

Text Book:

1. Forouzan, B. A.: Data Communications and Networking, 2/Ed (TMH)

Reference Books:

1. Tanenbaum, A. S.: Computer Networks, 4/Ed (PHI)

CS. 2.2 ADVANCED JAVA

OBJECTIVES

- To know the fundamentals of Java programming and develop error-free, well-documented Java programs
- To develop and test various advanced concepts of Java such as Java network, search engine, and web framework programs.
- Learn how to write, test, and debug advanced-level Object-Oriented programs using Java.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Know some concepts of advanced programming and practice on reusing components.
- Write sophisticated Java applications.
- Use the Java language for writing well-organized, complex computer programs with both command line and graphical user interfaces.

UNIT-I

Introduction to JAVA & its various features, JAVA Virtual Machine its architecture. Installation of JDK and 'CLASSPATH' setting, A First Java Program, Compilation and Applications, The JDK Directory Structure ,Lexical issues of java Class, Object, Instance Data and Class Data, Methods, Constructors, Access Modifiers, Destroying Objects , inheritance, overriding , Dynamic method dispatch abstract class interface ,Wrapper class boxing unboxing autoboxing and autounboxing, Package, multithreading , exception handling ., console and File I/O

UNIT-II

GUI basic, introduction to swing difference between AWT and swing , Swing components and containers Layout managers, event handling , Applets ,life cycle of applets steps for making applet, JLabel,JButton, JCheckBox, JRadioButton, JScrollPane, JTextField , JTextArea ,JMenu, JTable ,dialog boxes.

UNIT-III

JDBC concept The JDBC Connectivity Model, JDBC drivers ,Database Programming, Connecting to Database, Working with database tables, SQLWarning Classes, Executing SQL Queries, ResultSet MetaData, PreparedStatement, Parameterized Statements, Stored Procedures and Transaction Management, Networking , Basics of Networking, Inet Address, TCP/IP Sockets ,Data Grams, Simple Client Server socket programming. Remote method invocation (RMI)

UNIT-IV

J2EE Overview, Client Tier, Middle Tier, Application Server Tier, The J2EE Platform, Servlet , life cycle of servlet steps for making servlet, deployment ,Deployment descriptor and its configuration , Session tracking The JSP Solution, JSP Syntax & Deployment, Variables and Expressions, Sessions in JSP, page and taglib Directives .

UNIT-V

Enterprise java beans(EJB) ,EJB architecture , Classification of EJB, Session Beans , Stateless and Stateful Session bean ,Bean class , Developing and running bean application ,MVC (Model View Control) architecture

JAR Concepts, Steps for creating simple jar files, Creating executable JAR Files.

Books:

1. JAVA The Complete Reference Herbert Schildt Tata McGraw-Hill
2. JAVA Server Programming Balck Book Kogent Dreamtech publication
3. Programming in JAVA Sachin Malhotra Saurabh Choudhury Oxford publication
4. Introduction to Java Programming Y. Daniel Liang Person publication

CS.2.3 OPERATING SYSTEM DESIGN

OBJECTIVES

- To understand Operating system structure and services.
- To understand the concept of a Process, memory, storage and I/O management.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify the low-level structure and internal mechanism of operating system.
- Understanding the performance and design trade-offs in complex software systems.
- Describe the main responsibilities of a contemporary operating system (OS).
- List the most fundamental subsystems of an OS and the functions that each subsystem is responsible.
- Recognize and give examples of conflicting goals and compromises necessary in implementing an OS and configuring its run-time parameters

UNIT- I

Operating System, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special Purpose Systems, Computing Environments, Open-Source Operating Systems. Operating System Services, User Operating System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating System Structure, Virtual Machines, Operating System Debugging, Operating System Generations. System Boot.

UNIT- II

Process: Process Concept, Process Scheduling, Operations on Processes, Inter-Process Communication, Examples of IPC Systems, Communication in Client-Server Systems. Multithreaded Programming: Multithreading Models, Thread Libraries, Threading Issues, Operating-System Examples.

UNIT- III

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling. Multiple-Process Scheduling. Synchronization: The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Monitors, Synchronization Examples, Atomic Transactions.

UNIT- IV

Deadlocks: System Model, Deadlock Characterization, Methods of Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, Recovery from Deadlock. Memory Management Strategies: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium.

UNIT- V

Virtual-Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory. File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection.

TEXT BOOK:

Operating System Concepts**:** Silberschatz, Galvin, Gagne, 8/e (Wiley-India)

CS.2.4 THEORY OF COMPUTATION

OBJECTIVES

- To learn the mathematical foundations of computation including automata theory
- To learn the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.
- To learn about how really computers works and what kind of activities can be computed mechanically within a computer.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Model, compare and analyse different computational models using combinatorial methods.
- Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
- Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.
- Identify limitations of some computational models and possible methods of proving them.
- Have an overview of how the theoretical study in this course is applicable of application like designing the compilers.

UNIT I: - Regular Languages & Finite Automata:

Deterministic Finite Automata, Non-deterministic Finite Automata, Equivalence of NFA, and DFA. Regular Expressions & Languages, Conversion of DFAs to Regular Expressions, and vice versa. Properties of Regular Languages: Pumping Lemma, Closure properties: Union, Intersection, Complement, Difference, Reversal, Homomorphism, and Inverse Homomorphism. Decision Problems for Regular Languages, DFA Minimization.

UNIT II: - Context Free languages & Pushdown Automata

Context Free Languages, Context Free Grammars, Derivation, Ambiguity, Parsing. Pushdown Automata: Definition of PDAs, Acceptance of PDAs by final state and by empty stack. Conversion of CFG to PDA and vice versa. DPDAs & DCFLs, Determinism & Parsing. Simplification of CFG's, Chomsky Normal Form. The Pumping Lemma for CFL's. Closure properties: union, concatenation, *, +, Homomorphisms, and Reversal. Nonclosure under reversal and complementation. Decision Problems for CFLs, CYK Algorithm, Undecidable Problems for CFLs.

UNIT III: - Turing Machines:

TM Definition and Notation; Instantaneous Descriptions, NTM & DTM, Programming tricks for TMs, Examples involving TM Computations, Extensions & Restrictions to Basic TM Model, (Multi Tape, Multi Dimensional, Counter machine, Two Stack PDAs).

UNIT IV: - Decidability Theory:

The Church-Turing Thesis, Universal Turing Machines and TM Encoding. Decidable and semi-decidable languages, Recursive Enumeration and Decidability, Many-one Reductions, Hardness, Undecidability, Closure Properties. The Diagonalization Language, The Halting Problem, Post's Correspondence Problem, Undecidable Problems from Language Theory, Rice's Theorem. Linear Bounded Automata (LBA).

UNIT V: - Complexity Theory

Measuring Complexity, The Big **Oh**, **Theta, Big Omega** Notations, Time Complexity classes: P, NP, NP-Completeness, Coping with NP-Completeness. Cook-Levin's Theorem, Some NP-Complete Problems: SAT, 3-SAT, Hamiltonian Path, Vertex Cover, Independent Set. Space Complexity classes: PSPACE, L, NL.

Text Book: Introduction to Automata Theory, Language & Computation-Hopcroft, Motwani and Ullman

CS.2.5 DATA MINING

OBJECTIVES

- To introduce students to the basic concepts and techniques of Data Mining
- To develop skills of using recent data mining software for solving practical problems.
- To study the methodology of engineering legacy databases for data warehousing and data mining to derive business rules for decision support systems
- Develop and apply critical thinking, problem-solving, and decision-making skills which can initiate students about research oriented thinking.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Understand what data mining is all about.
- Perform the data preparation tasks and understand the implications.
- Demonstrate an understanding of the alternative knowledge representations such as rules, decision trees, decision tables, and Bayesian networks.
- Demonstrate an understanding of the basic machine learning algorithmic methods that support knowledge discovery.
- Identify alternative data mining implementations and what might be most appropriate for a given data mining task.

UNIT-I

Introduction: Definition of data mining-data mining vs query tools-machine learning-taxonomy of data mining tasks - steps in data mining process - overview of data mining techniques.

UNIT-II

Data Pre-Processing And Characterization: Data Cleaning - Data Integration and Transformation - Data Reduction - Discretization and Concept Hierarchy Generation - Primitives - Data Mining-Query Language- Generalization-Summarization-Analytical Characterization and Comparison

UNIT-III

Association Rule - Mining: Market basket analysis, frequent Itemset generations, The Apriori principle, Candidate Itemset generation and Pruning, Support counting using Hash tree, Multi Dimensional data from Transactional Database and Relational Database. FP-Growth Algorithm, objective measures of Interestingness

UNIT-IV

Classification: Classification - Decision Tree Induction - Bayesian Classification - Back Propagation , Lazy learners, nearest neighbor, Rule based classification, Accuracy, Prediction-Linear regression, Non-linear regression models

UNIT-V

Cluster analysis: Types of data, Distance measures, Evaluation criteria measures, Clustering Methods - Partitioning methods, K-Means, Density based method- DBSCAN, Model based clustering methods – Expectation-maximization, outlier analysis.

Text Books

1. Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufman Publishers, 2006.

Reference Books

1. Usama M.Fayyad, Gregory Piatetsky Shapiro, Padhrai Smyth, Ramasamy Uthurusamy, Advances in Knowledge Discover and Data Mining, The M.I.T.Press, 2007.
2. Ralph Kimball, Margy Ross, The Data Warehouse Toolkit, John Wiley and Sons Inc., 2002.
3. Alex Berson, Stephen Smith, Kurt Thearling, Building Data Mining Applications for CRM, Tata McGraw Hill, 2000.
4. Margaret Dunham, Data Mining: Introductory and Advanced Topics, Prentice Hall, 2002.
5. Daniel T. Larose John Wiley & Sons, Hoboken, Discovering Knowledge in Data: An Introduction to Data Mining, New Jersey, 2004.
6. M.Panda, S.Dehuri and M.R.Patra, Modern Approaches of Data Mining: Concepts and techniques, Narosa Publications, 2016

CS.2.5 (b) COMPUTER GRAPHICS

OBJECTIVES

The basic objective of the course is-

- To understand the concept of representation of information in graphical and pictorial way.
- Different types display of information by computer.
- Two dimension and three-dimension transformation and translation.

LEARNING OUTCOMES

Upon successful completion of this course, the student shall be able to:

- Understand the applications of computer graphics, display device and operating principle.
- Identify the different Line, circle and ellipse drawing Algorithms.
- Analyze the 2D transformation, clipping.
- Demonstrate 3D transformation and viewing.
- Classify the visible surface detection methods, Illuminating models.

UNIT-I

A survey of computer graphics: Computer Aided Design, Presentation Graphics, Computer Art, Entertainment, Education Training, Visualization, Image Processing, Graphical User Interface. Overview of graphics system: Video Display Devices, Raster Scan Systems, Random Scan Systems, Input Devices, Hard-copy Devices, Graphics Software. Output primitives: Points and lines, DDA and Bresenham's Line Drawing Algorithm, Midpoint circle algorithm, Filled area primitives. Attributes of output primitives: Line attributes, Curve attributes, Colour and grayscale levels, Area-fill attributes, Character attributes, Bundled attributes.

UNIT-II

Two dimensional geometric transformations: Basic Transformation (Translation, Rotation, Scaling), Matrix representation and homogenous coordination, Composite Transformation, Reflection, Shear. Two dimensional viewing: The viewing Pipe-line, Viewing Coordinate Reference frame, Window-to-viewport coordinate transformation. Clipping: Line Clipping (Cohen Sutherland Algorithm), Polygon clipping (Sutherland-Hodgemen Algorithm)

UNIT-III

Three dimensional object representation: Polygon Surfaces, Quadratic surfaces, Spline Representations, Beizer Curves and surfaces, B-Spline Curves and surfaces

UNIT-IV

Three dimensional geometric and modeling transformations: Translation, Rotation, Scaling, Reflections, Shears, Composite Transformation. Three dimensional viewing: Viewing pipeline, Viewing coordinates, Projections (Parallel and Perspective).

UNIT-V

Visible-surface detection methods: Classification of visible-surface detection algorithms, back-face detection, Depth-Buffer method, A-Buffer method, Scan-line method, Depth- sorting method. Illumination Models: Basic illumination models, Displaying light intensities, Halftone Patterns and Dithering Technique, Polygon Rendering Methods, (Gouraud and phong shading)

Text books:

1. Computer Graphics C Version, by D. Heam and M. P. Baker, 2nd Edition, Pearson Education, 2002.

Reference books:

1. Marschner, S., & Shirley, P. (2018). Fundamentals of computer graphics. CRC Press.
2. Foley, J. D., Van Dam, A., Feiner, S. K., Hughes, J. F., & Phillips, R. L. (1994). Introduction to computer graphics (Vol. 55). Reading: Addison-Wesley

CS.3.1. ARTIFICIAL INTELLIGENCE

OBJECTIVES

- To learn the basic concepts of AI principles and approaches.
- To develop the basic understanding of the building blocks of AI.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Have fundamental understanding of the basic concepts of artificial intelligence (AI).
- Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
- Have fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Have knowledge of current scope and limitations, and societal implications of AI.
- Have basic foundation of machine learning.

UNIT-I

Introduction to AI , History of AI , State of Art Intelligent Agents, Problem Solving by Searching : BFS, Uniform Cost Search, DFS, IDS, Bi-directional Search, Constraint Satisfactory Search, Informed Search Best First Search, Heuristic Function, Memory bounded search, A* and IDA*, Game Playing: Min-Max search and Alpha-Beta pruning.

UNIT-II

Knowledge & Reasoning : Agents that reason logically, First Order Logic, Syntax and Semantics. Inference in First Order Logic : Inference Rules, Modus Ponems, Unification, Forward and Backward Reasoning, Resolutions Planning : A simple Planning Agent, from Problem Solving to Planning, Planning in Situation Calculus.

UNIT-III

Learning : Learning from Observations . A General Model of Learning Agents, Inductive Learning; Expert Systems, Architecture, Knowledge Acquisition, MYCIN ; Natural Language Processing : Syntactic Processing, Semantic Analysis, Efficient parsing.

UNIT-IV

Introduction to Pattern Recognition: Recognition & Classification Process, learning, Classification Patterns, Visual Image Understanding, Image Transformation; Perception: Image Formation, Image Processing Operations for easy Vision, Speech, Recognition. Introduction to Robotics.

UNIT-V

Prolog Programming : Basic Prolog Concepts, Facts, Rules, Structures, Lists, Executing and meaning of Prolog Programs, Recursive Programming, Backtracking with cuts.

Text Book:

Stuart Russel &, Peter Norvig: Artificial Intelligence A Modern Approach (Person Education Asia.) 3rd edition.

CS.3.2 SOFTWARE ENGINEERING

OBJECTIVES

- To learn the way of developing software with high quality and the relevant techniques.
- To introduce software engineering principles for industry standard.
- To focus on Project management domain and Software risks management.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify, formulate, and solve complex problems by applying principles different principles of software engineering.
- Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
- Communicate effectively with a range of audiences and recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
- Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

UNIT-I

Computer-Based System Engineering: Emergent System Properties, Systems and their Environment, System Modeling, System Engineering Process, System Procurement. Software Processes: Software Process Models, Process Iteration, Software Specification, Design and Implementation, Software Validation and Evaluation, Automated Process Support. Project Management: Management Activities, Project Planning, Project Scheduling, Risk Management.

UNIT-II

Software Requirements: Functional and Non-Functional Requirements, User Requirements, System Requirements, Software Requirements Document. Requirements Engineering Processes: Feasibility Studies, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management. System Models: Context Models, Behavioral Models, Data Models, Object Models, CASE Workbenches.

UNIT-III

Architectural Design: System Structuring, Control Models, Modular Decomposition, Cohesion and Coupling, Data Flow-oriented design. Distributed System Architectures: Multiprocessor Architectures, Client-Server Architectures, Distributed Object Architectures, CORBA. Object-Oriented design. Real-Time Software Design: System Design, Real-Time Executives, Monitoring and Control Systems, Data Acquisition Systems. Design with Reuse: Component-Based Development, Application Families, Design Patterns.

UNIT-IV

Verification and Validation: Verification and Validation Planning, Software Inspections, Automated Static Analysis, Clean-room Software Development. Software Testing: Defect Testing, Integration Testing, Object-Oriented Testing, Testing Workbenches. Software Cost

Estimation: Productivity, Estimation Techniques, Algorithmic Cost Modeling, Project Duration and Staffing.

UNIT-V

Dependability: Critical Systems, Availability and Reliability, Safety, Security. Critical Systems Specifications: Software Reliability Specification, Safety Specification, Security Specification. Critical Systems Development: Fault Minimization, Fault Tolerance, Fault Tolerance Architectures, Safe System Design.

Text Book:

Sommerville, I: Software Engineering, 6/e

Reference Book

1. Pressman, R. S: Software Engineering, 4/e (McGRAW-HILL)
2. Aggarwal, K. K. & Singh, Y: Software Engineering (New Age International)

CS.3.3 COMPILER DESIGN

OBJECTIVES

- To provide a thorough understanding of the internals of Compiler Design.
- To explore the principles, algorithms, and data structures involved in the design and construction of compilers.
- Topics include context-free grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation, and code optimization.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Realize basics of compiler design and apply for real time applications.
- Introduce different translation languages
- Understand the importance of code optimization
- Know about compiler generation tools and techniques
- Working of compiler and non-compiler applications
- Compiler for a simple programming language

UNIT–I

Compilers & Translators, Need of Translators, Structure of a Compiler, Phases, Lexical Analysis, Syntax Analysis, Intermediate Code Generation, Code Optimization, Code Generation, Book Keeping, A Symbol Table in brief, Semantic Analysis, L-value, r-values, Error Handling.

UNIT–II

Rules of Lexical Analyser, Need for Lexical Analysis, Input Buffering, Preliminary Scanning, A simple Approach to the Design of Lexical Analysers, Transition Diagrams, Regular Expression, String & Languages, Finite Automata, Non-deterministic Automata, Deterministic Automata, From regular Expression to Finite Automata, Context free Grammars, Derivations & Parse Trees, Parsers, Shift Reduce Parsing, Operator- Precedence Parsing.

UNIT–III

Symbol Table Management, Contents of a Symbol Table, Names & Symbol table records, reusing of symbol table spaces, array names, Indirection in Symbol Table entries, Data Structures for Symbol Tables , List, Self Organizing Lists, Search Trees, Hash Tables, Errors, Reporting Errors, Sources of Errors Syntactic Errors, Semantic Errors, Dynamic Errors, Lexical Phase Errors, Minimum Distance Matching, Syntactic Phase Error, Time of Detection, Ponic mode, Case study on Lex and Yacc.

UNIT–IV

Principal Sources of Optimization, Inner Loops, Language Implementation Details Inaccessible to the User. Further Optimization, Algorithm Optimization, Loop Optimization , Code Motion, Induction Variables, Reduction in Strength, Basic Blocks, Flow Graphs, DAG Representation of Basic Blocks, Value Numbers & Algebraic Laws, Global Data Flow Analysis, Memory Management Strategies , Fetch Strategy, Placement Strategies, Replacement Strategies, Address Binding, Compile Time, Load Time, Execution Time, Static Loading, Dynamic Loading, Dynamic Linking.

UNIT–V

Problems in Code Generation, a Simple Code Generator, Next-Use Information, Register Descriptors, Address Descriptors, Code Generation Algorithm, Register Allocation & Assignment, Global Register Allocation, Usage Counts, Register Assignment for Outer Loops, Register Allocation by Graph Coloring , Code Generation from DAG's, Peep-Hole Optimization, Redundant Loads & Stores, Un-Reachable Code, Multiple Jumps, Algebraic Simplifications, Use of Machine Idioms.

Text Book:

Compilers, Techniques and Tools (2^{nd} edition), A.V.Aho, M.S.Lam, Ravi Esthi and J.D.Ullman

CS.3.4 (a) Cryptography and Network Security

OBJECTIVES

The basic objective of the course is-

- Student will be able to understand basic cryptographic a algorithms, message and web authentication and security issues.
- Ability to identify information system requirements for both of them such as client and server.
- Ability to understand the current legal issues towards information security.

LEARNING OUTCOMES

Upon successful completion of this course, the student shall be able to:

- Understand cryptography and network security concepts and application
- Apply security principles to system design
- Identify and investigate network security threat
- Analyze and design network security protocols
- Conduct research in network security

UNIT-I

Introduction : Introduction to Cryptography, Security Threats, Vulnerability, Active and Passive attacks, Security services and mechanism, Conventional Encryption Model, CIA model

Math Background : Modular Arithmetic, Euclidean and Extended Euclidean algorithm, Prime numbers, Fermat and Euler's Theorem

UNIT-II

Classical Cryptography : Dimensions of Cryptography, Classical Cryptographic Techniques

Block Ciphers (DES, AES) : Feistal Cipher Structure, Simplifies DES, DES, Double and Triple DES, Block Cipher design Principles, AES, Modes of Operations

UNIT-III

Public-Key Cryptography : Principles Of Public-Key Cryptography, RSA Algorithm, Key Management, Diffie- Hellman Key Exchange, Elgamal Algorithm, Elliptic Curve Cryptography

UNIT-IV

Hash and MAC Algorithms : Authentication Requirement, Functions, Message Authentication Code, Hash Functions, Security Of Hash Functions And Macs, MD5 Message Digest Algorithm, Secure Hash Algorithm, Digital Signatures

Key Management : Key Distribution Techniques, Kerberos

UNIT-V

Security in Networks : Threats in networks, Network Security Controls – Architecture, Encryption, Content Integrity, Strong Authentication, Access Controls, Wireless Security, Honeypots, Traffic flow security, Firewalls – Design and Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP, S/MIME

Text books:

1. Cryptography and Network Security : William Stallings, Pearson Education,4"' Edition
2. Cryptography and Network Security : Atul Kahate, Mc Graw Hill Edition

Reference books:

1. Wade Trappe, Lawrence C Washington, " Introduction to Cryptography with coding theory", Pearson.
2. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.
3. Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in computing – Prentice Hall of India.

CS.3.5 (b) Machine Learning

OBJECTIVES

The objective of the course is to study and analyze data and their interpretation. Ultimately processing the data in an intellectual way to achieve business requirement. At the end of the course, the students will be able to:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

LEARNING OUTCOMES

Upon successful completion of this course, the student shall be able to understand:

- Decision tree learning algorithm.
- Neural network, hypothesis accuracy estimation.
- Supervised Learning to obtain a predicted output.
- Unsupervised Learning on data.

UNIT-I

Supervised Learning 1: Overview of supervised learning, classification, and regression problems, K-nearest neighbourhood (KNN) classifier, variation of k-NN classifiers. Decision tree learning, Issues in Decision tree learning. Linear regression, Multiple linear regression, Logistic regression.

UNIT-II

Model Assessment and Selection: Loss function, test and training error, Bias, Variance, and model complexity, Bias-variance trade off, Bayesian approach and BIC, Cross- validation, Boot strap methods, Performance of Classification algorithms (Confusion Matrix, Precision, Recall and ROC Curve).

UNIT-III

Support Vector Machines (SVM) and Neural network: SVM for classification, Reproducing Kernels, SVM for regression. Model of a neuron, Perceptron learning, Backpropagation, Methods to improve backpropagation.

UNIT-IV

Boosting and Bayesian Learning: Boosting methods (AdaBoost, gradient boosting, XG boost), Bayes Theorem and concept learning, Maximum likelihood and least- squared error hypotheses, Bayes optimal Classifier, Naïve Bayes Classifier.

UNIT-V

Unsupervised Learning and Random forests: Cluster analysis (k-means, Hierarchical clustering, DBSCAN, spectral clustering), Gaussian mixtures and EM algorithm, Random forests and analysis. Feature Extraction (Principal Component Analysis (PCA).

Text books:

1. Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition, 2013
2. Alpaydin, E. , Introduction to Machine Learning. United Kingdom: MIT Press (2014).

Reference books:

1. C. M. Bishop –Pattern Recognition and Machine Learning, Springer, 2006
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman *, The Elements of Statistical Learning-Data Mining, Inference, and Prediction*, Second Edition , Springer Verlag, 2009.