CURRICULA-2020

MCA

(2 Years - CBCS Scheme)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OFNATURAL SCIENCES

Jamia Millia Islamia: New Delhi, India

Recommended by the Board of studies DCS-FNS-JMI on 26/08/2020
Detailed Syllabi approved by the BOS DCS-FNS-JMI on 20/01/2021

MCA (2 Years): 2020

A.	**Reference**	1. Notification No.AICTE/AB/MCA/2020-21/03-07020) 2. Adaptation by BOS dated 06-07-20 for MCA(2-Years) program.
B.	**Motivation**	1. Rapidly changing academic and technological scenarios around the world. 2. Conformance to prescriptions from the regulatory bodies, and JMI ordinances. 3. Keeping abreast with the current and future industrial skill-set requirements, and placement related tests.
C.	**Bases and Constraints**	UGC Guideline on Choice Based Credit Based System (CBCS) – with greater autonomy to students on the selection of courses – by reinforcing the following: 1. Retaining prescribed types as Core, Electives, CBCS, Ability/Skill enhancement courses. 2. Retaining the 4thsemeater as full project-semester, because of the mandate of MCA as a professional program and placements. 3. CBCS courses are to be open for any PG students including DCS students, subject to availability of seats. 4. One CBCS-Elective and CBCS-Ability/Skill-Enhancement course each, to be compulsory in each of the semesters (1-3).
D.	**Abbreviations**	Computer Science: **CS**, Core: C, Elective: E, Ability/Skill: S
E.	**Course Codes**	Unique code to be assigned to each of the typical courses at (UG/PG level separately), offered by the Department using the following coding scheme: • Computer Science Core/elective: CSC/E • Computer Science (CBCS Elective): CBCSE • Computer Science (CBCS Ability/Skill-Enhancement):CBCSS
F.	**Course L-T-P**	1. Core Theory Courses: 3-0-0 2. Lab-Oriented Theory Courses (Electives): 3-0-2 3. Lab-Courses: 0-0-4 4. CBCS Elective Courses: 4-0-0/3-0-2 5. Ability/Skill Enhancement Courses: 2-0-2 6. Major Project: 0-4-32
G.	**Eligibility for Admission**	1. Bachelor’s Degree with at least 50% Marks in (a) Computer Science/Engineering/Applications/Equivalent Allied, OR (b) any other discipline with Mathematics at 10+2 Level. 2. In case of 1(b), a candidate must either produce ‘a valid certificate of passing at least 6 credits of Computer Science courses (with at least C grade/50% marks) from any Govt-approved mode at 10+2/graduation level’ **OR** ‘complete the bridge courses from the Department in the first semester’.
H.	**Bridge Courses**	Applicable to those qualifying under G1(b), under which a student must produce or complete at least 6 credits of the courses as advised by the department, necessarily including a course on programming.
I.	**Special Considerations**	• Accommodation of non-Computer Science graduates and the bridge-courses. • Applied discipline and professional nature of programs. • Balancing academic, technological, and industrial imperatives. • National and global connect. • ACM, UGC, AICTE, and other central universities' curriculum.
J.	**Track Specialization**	MCA degree may be awarded in a specialization (Advanced Computing OR Informatics), provided a student, besides fulfilling all the requirements: (i) opts all the electives from that track, and (ii) requests formally in writing to the Department.
K.	**Remarks**	1. Students are encouraged to enroll in courses about Communication Skills and Management from other Departments, as CBCS electives. 2. CBCS courses of minimum 3 credits each that may be chosen from other departments subject to students’ requirements and convenience. 3. Lab and project courses shall have independent practical and viva-voce examinations. 4. At least two courses shall be offered for each of the typical electives, provided at least 15 students opt for a course. 5. Faculty members may be allocated 2 to 4 lab-periods/week subject to feasibility. 6. Department may float any other elective, beyond the listed ones, subject to feasibility and endorsement of BOS.

MCA(2 Years) Programme Structure: 2020

SEM	CODE	COURSE TITLE	CREDITS	REMARKS
Bridge Courses	CSC01	Computer Fundamentals	3	Any two courses as advised by the Department, as per clause H.
	CSC02	Programming and Problem Solving using C	4	
	CSC03	Applied Operating Systems	3	
	CSC04	Information Systems	3	
I	CSC11	Digital Logic and Computer Architecture	3	Periods/Week: ~24 Credits: ~20
	CSC12	Discrete Mathematics	3	
	CSC13	Algorithmics and Program Design	3	
	CSC14	Database Management Systems	3	
	CSC15	Lab-I (APD)	2	
	CSC16	Lab-II (DBMS)	2	
	CBCSE17	CBCSE-I	4	
II	CSC21	Software Engineering	3	Periods/Week: ~31 Credits:~27
	CSC22	Object Oriented Programming	3	
	CSC23	Advanced Data Structures	3	
	CSC24	Operating Systems and Shell Programming	3	
	CSE25	Elective-I	4	
	CSC26	Lab-III (OOP)	2	
	CSC27	Lab-IV(ADS+SP)	2	
	CBCSE28	CBCSE-II	4	
	CBCSS29	CbcsS-I	3	
III	CSC31	Data Communication and Networks	3	Periods/Week: ~31 Credits: ~27
	CSC32	Artificial Intelligence	3	
	CSC33	Information Security	3	
	CSC34	Analysis and Design of Algorithms	3	
	CSE35	Elective-II	4	
	CSC36	Lab-V (AI)	2	
	CSC37	Lab-VI (ADA)	2	
	CBCSE38	CBCSE-III	4	
	CBCSS39	CbcsS-II	3	
IV	CSC41	Major Project	20	Periods/Week: 36 and Credits: 20
Summary: Core-Courses(12)+Elective(2)+Lab-Courses(6)+CBCS-Elective(3)+CBCS-Skill(2)+Major-Project (1)=26 Courses				

Bi-Track Specialization: Optional Model

Department Elective Courses for PG Students: L-T-P: 4-0-0 /3-0-2 (4 Credits)

Specialization Track		**Advanced Computing**	**Informatics**	
II	CSE25	Advanced DBMS Computer Graphics Theory of Computation	Software Quality Assurance Decision Support Systems Software Project Management	Management Information Systems Data Mining and Warehousing Software Testing and Verification
III	CSE35	Distributed Systems Cryptography Compiler Design System Software Digital Image Processing	Soft Computing Cyber Security Cloud Computing Machine Learning Security Audit and Design	Eduinformatics Health Informatics Advanced Software Engineering Adhoc Networks Programming with Java

CBCS Courses for PG Students: L-T-P: 4-0-0/3-0-2 (4 Credits)

I	CBCSE17	Statistical Computing	e-Business Systems	Human Computer Interaction
II	CBCSE28	Modelling and Simulation	Business Informatics	Social Informatics
III	CBCSE38	Numerical Optimization	IT Management	Multimedia Applications

CBCS (Ability/Skill Enhancement) Courses for PG Students: L-T-P: 2-0-2 (3 Credits)

II	CBCSS29	Programming with Python	Mobile Applications	Portal Development
III	CBCSS39	Data Analytics with HADOOP	MATLAB Computations	Web Based Programming

NOTE:- Relevant CBCS courses, of minimum 3 credits, may also be chosen from other departments.

Detailed – Syllabi

(Remark: Updated syllabus of some courses shall be designed as and when floated for teaching.)

Section-1: Core Courses

CSC11: Digital Logic and Computer Architecture

LEARNING OUTCOMES

Understanding of Boolean Algebra and Simplification of Boolean Functions
Understanding the digital logic gates, Combinational Logic and Sequential Logic
Understanding of Microprocessor Architecture and Micro-operations
Understanding of CPU and Binary Arithmetic

1. **Information Representation:** Number Systems - Binary, Octal, Decimal, and Hexa-Decimal; Number Base Conversions; Binary Arithmetic; Complements: (r-1)'s Complement, r's Complement, Subtraction using Complements; Floating Number-Fixed-point Representation, Floating-point Representation; Binary Codes for Decimal Digits: BCD Code, Excess-3 Code, 84-2-1 Code, 2421 Code, Reflected Code; Error Detection Code; Character Representation – ASCII, EBCDIC.
2. **Boolean Algebra, Logic Gates and simplification**: Boolean Algebra-Basic Definitions, Postulate, Basic Theorems and Properties of Boolean Algebra; Boolean Functions, Canonical and Standard Forms: Minterms and Maxterms, SOP, POS Conversion Between Canonical Forms, Standard Form of a Boolean Function; Other Logical Operations; Digital Logic Gates, Implementation of Boolean Functions, Simplification using boolean Algebra and Karnaugh Maps (K-Map) Method.
3. **Cobinational and Sequential Logic Circuit:** Overview of Combinational Logic; Combinational Logic Design Procedure; Design of Some Standard Combinational Circuits: Half Adder, Full Adder, Half Subtractor, Full Subtractor, Code Conversion; Decimal Adder, BCD Adder, Magnitude Comparator, Decoders, Encoder, Multiplexers, De-multiplexer, Flip-Flops: RS Flip Flop, Clocked RS, JK Flip Flop, Master Slave JK Flip Flop, D Type Flip Flop, T Type Flip Flop, State Table, State Diagram, State Equations, Flip Flop Characteristic Tables; Flip Flop Excitation Tables; Design of Sequential Circuits.
4. **Register Transfer and Micro Operations:** Register Transfer Language (RTL); Register Transfer; Bus Transfer; Memory Transfers; Arithmetic Microoperations; Logic Microoperations, List of Logic Microoperations, Addressing Modes, Data Transfer.
5. **Central Processing Unit (CPU):** Introduction; General Register Organization; Control Word; Stack Organization – Register Stack, Memory Stack, Reverse Polish Notation, Evaluation of Arithmetic Expression. Instruction Format – Three Address Instructions, Two Address Instructions, One Address Instructions, Zero Address Instructions. Parallel Processing; Pipelining – Arithmetic Pipeline, Instruction Pipeline,

REFERENCES

Mansaf Alam & Bashir Alam: Digital Logic Design. PHI
M. Morris Mano: Digital Logic and Computer Design. Pearson
M. Morris Mano: Computer System Architecture. Pearson
William Stalling: Computer Organization and Architecture. Prentice Hall
V. Rajaraman & T. Radhakrishnan: Computer Organization and Architecture. PHI
Donald D. Givone: Digital Principles and Design. McGraw Hill

CSC12: Discrete Mathematics

LEARNING OUTCOMES

Comprehend the values of basic structures in computer science and their algorithmic utility & significance.
Illustrate and apply logic propositional and predicate logic in decision problems.
Appreciate the context and utility of relations and posets in computer science.
Solve practical problems of complexity analysis by counting.
Illustrate the typical graph techniques and their applications.

1. **Basic Structures**: Set, Multi-set and Sequences; Type of sets, Set Operations, Power Set, Cartesian Products, Relation, Representation of relation, composition of relations, Functions, Types of Functions, Inverse of a functions, Compositions of functions, function representation, Sequences, Special Integer Sequences, Summations.
2. **Relations and Partial Orders:** Equivalence Relation, Reflexive, Symmetric and Transitive Closure, Transitive Closure and Warshall's Algorithm; Equivalence Classes and Partitions; Partial Ordering, Lexicographic Order, Hasse Diagram, Maximal and Minimal Elements, Lattices; Algebraic structure: Semi group, monoid.
3. **Logic, Reasoning and Inferences:** Foundations of Logic; Propositions, Conditional, Bi-conditionals, Truth Table, Precedence of Logical Operators, Translating English Sentences, System Specifications, Logic Puzzles, Propositional Logical, De Morgan's Laws, New Logical Equivalences; Predicates and Quantifiers – Predicates, Quantifiers, Universal Quantifiers, Quantifiers with Restricted Domains, Precedence, Binding Variables, Logical Equivalences, Negating Quantified Expressions, Translation to English Expressions, Rules of Inferences, valid Arguments, Proof Techniques.
4. **Counting:** Simple and Complex Counting Problems, Inclusion-Exclusion Principle; Tree Diagrams; The Pigeonhole Principle; Permutations, Combinations, Binomial Coefficients, Examples and Applications; Binomial Coefficients, Binomial Theorem, Expression, and other Identities; Permutations and Combinations with Repetition, Permutations with Indistinguishable Objects, Distributing Events into Boxes; Generating Permutations and Combinations.
5. **Recurrences:** homogeneous linear recurrences, Non-homogeneous linear recurrences, Solving recurrences using induction method, Solving recurrences using characteristic equations, Solving recurrences by domain transformation, Solving recurrences by substitution.

REFERENCES

K. Rosen: Discrete Mathematics and its Applications with Combinatorics and Graph Theory. McGraw Hill
Bernard Kolman, Robert Busby, and Sharon C. Ross: Discrete Mathematical Structures. Prentice Hall
J.P. Tremblay, and R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill

CSC13: Algorithmics and Program Design

LEARNING OUTCOMES

Illustrate algorithmic terminology and issues; and analyze the efficiency of algorithms.
Develop iterative/recursive flowcharts and algorithms for basic computational problems.
Devise factoring algorithms, analyze and develop their improved versions.
Implement and analyze different array-based searching and sorting algorithms.
Design well-styled programs, trace and test the same.

1. **Algorithmic Problem Solving**: Algorithms; Problem Solving Aspect: Algorithm Devising, Design and Top-down Design; Algorithm Implementation: Essential and Desirable Features of an Algorithm; Efficiency of an Algorithm, Analysis of Algorithms, Pseudocodes; Algorithm Efficiency, Analysis and Order; Importance of Developing Efficient Algorithms; Complexity Analysis of Algorithms: Every-Case Time Complexity, Worst-Case Time Complexity, Average-Case Time Complexity, Best-Case Time Complexity, Flowchart. Flowchart – Symbols and Conventions.
2. **Basic Algorithms** – Exchanging the Values of Two Variables, Counting, Summation of a Set of Numbers, Factorial Computation, Sine Function Computation, Generation of the Fibonacci Sequence, Reversing the Digits of an Integer, Base Conversion, etc., Recursive Algorithms.
3. **Factoring:** Finding the square root of number, Smallest Divisor of an integer, Greatest common divisor of two integers, generating prime numbers, computing prime factors of an integer, Generation of pseudo random numbers, Raising a number to a large power, Computing the *n*th Fibonacci number.
4. **Arrays, Searching and Sorting:** Single and Multidimensional Arrays, Array Order Reversal**,** Array counting, Finding the maximum/minimum number in a list, Efficient algorithm for finding max-min in a list, partitioning an array, Monotones Subsequence; Searching: Linear and Binary Array Search, interpolation search; Sorting: Sorting by selection, Exchange and Insertion. Sorting by diminishing increment, Sorting by partitioning.
5. **Programming:** Introduction, Game of Life, Programming Style: Names, Documentation and Format, Refinement and Modularity; Coding, Testing and Further Refinement: Stubs and Drivers; Program Tracing, Testing, Evaluation; Program Maintenance: Program Evaluation, Review, Revision and Redevelopment; and Problem Analysis, Requirements Specification, Coding and Programming Principles.

REFERENCES

R.G. Dromy: How to Solve by Computer. Pearson (Unit 1-4)
R. Kruse, C.L. Tondo, B. Leung, and S. Mogalla: Data Structures and Program Design in C. Pearson (Unit-5)
L.A. Robertson: Simple Program Design, A Step-by-Step Approach. Thomson

CSC14: Database Management System

LEARNING OUTCOMES

Fundamentals of databases with their advantages and functionalities
Various database models. Details of RDBMS, its advantages and usage
ER Diagrams and EER Diagrams
Normalization and it various usage in table optimization

1. **Basic Concepts:** Data, Database and DBMS; Database vs. Traditional File System Approach; Three Schema Architecture of DBMS, Data Independence; Categories of Database Management Systems: Hierarchical, Network and Relational Database Systems.
2. **Database Models**: Introduction, Categories of Database Models: High-level or Conceptual Data Models, Representational or Implementation Data Models, Low-level or Physical Data Models, Object Data Models. Entity relationship (ER) Model: Basic Concepts and their representations – Entity, Entity Type and Entity Set, Attributes and Keys, Relationships, Relationship Types, and Structural Constraints, Weak Entity, Naming Conventions & Design Issues in ER Model. ER and EER Diagrams.
3. **Relational Database Model:** Structure of Relational Model; Domains, Attributes, Tuples, and Relations; Characteristics of Relations; Relational Constraints – Domain Constraints, Key Constraints, Entity Integrity, and Referential Integrity Constraints; Relational Database Schema; Relational Algebra Operations – Select, Project, Rename, Union, Intersection, Set Difference, Join, and Division Operations; Aggregate Functions and Groupings.
4. **Structured Query Language (SQL):** Schema, Table and Domain Creation; Schema and Table Deletion; Table Modification; Insert, Delete, and Update Statements; SELECT- FROM-WHERE Structure; Renaming Attributes; Nested Queries and Set Comparisons; EXISTS and UNIQUE Functions; Aggregate Functions; Creating and Updating Views. Introduction to PL/SQL.
5. **Functional Dependencies and Normalization:** Informal Design Guidelines for Relation Schemas; Functional Dependencies; Inference Rules for Functional Dependencies; Normalization using Functional Dependencies – First Normal Form (INF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF); Multi- Valued Dependencies and Fourth Normal Form (4NF); Join Dependencies and Fifth Normal Form (5NF); Relation Decomposition and Insufficiency of Normal Forms; Dependency Preserving and Lossless Join Decompositions; Null Values and Dangling Tuples. Transaction Management and Concurrency Control: Transaction Concept; Transaction State; Concurrent Executions; Serializability and Recoverability; Testing for Serializability. Concurrency Control – Lock-Based Protocols and Timestamp-Based Protocols.

REFERENCES

R. Elmasri, S. B. Navathe: Fundamentals of Database Systems. Pearson
A. Silberschatz, H. F. Korth, and S. Sudarshan: Database System Concepts. McGraw Hill
J. Casteel: ORACLE 9i Developer: PL/SQL Programming. Thomson
Ivan Bayross: SQL, PL/SQL The Programming Language of of Oracle. BPB

CSC15: Lab-I (APD)

Implementation of at least ONE specific assignment concerning each of the following:

1. Derive the worst case, best case and average case time complexity for searching and sorting algorithms.
2. Generate the Fibonacci series using iterative and recursive functions.
3. Implement Euclids division algorithms for calculating GCD.
4. Implement an algorithm to generate pseudo random numbers.
5. Implement merge and Quick sort algorithms and determine the time required to sort the elements.
6. Implement binary search to find an element in a sorted array.
7. Design and implement an algorithm to search for the maximum and minimum element in an array.
8. Implement interpolation searching.

CSC16: Lab-II (DBMS)

Implementation of at least ONE specific assignment concerning each of the following:

1. SQL statements to create, update, and delete databases, tables, and views.
2. SQL statements to insert, update, and delete records from tables.
3. Simple SQL queries to retrieve information from a database.
4. Nested SQL queries to handle complex information retrieval requirements.
5. Managing changes affecting the data using COMMIT, ROLLBACK and SAVEPOINT.
6. Providing security to databases using GRANT and REVOKE commands.
7. SQL queries using order by, group by and having clauses and SQL sub queries, nested queries, join operations.
8. PL/SQL blocks using basic data types, branching and looping constructs.
9. Database triggers, functions/procedures and packages using PL/SQL.

CSC21: Software Engineering

LEARNING OUTCOMES

Analyse and specify software requirements, and model its software design.
Understand the software life cycle models, and suitable model for the problem.
Illustrate design concepts and models and use suitable methods for software design

1. **Software Process:** Software Engineering and Development, Software and its Components; Software characteristics; problem of Size and Complexity; Evolving Role of software; Changing Nature; Legacy Software and Software Myths; Software Engineering – A Layered approach, Process Framework, CMMI; Technology, Product and Process.
2. **Software Process Models:** Prescriptive Models: Waterfall Model; Incremental Process Models , RAD; Evolutionary Models – Prototyping, Spiral, and concurrent Models; The Unified Process – Phases and Work Products; Agile Process Models – Extreme Programming and Adaptive; and Dynamic Software Development – Scrum, Crystal, Feature Driven and Agile Modeling.
3. **SE Principles and Practices:** Software Engineering Practices – Essence and Principles; Communication Practices; Planning Process; Modelling Principles; Construction Practices – Coding principles and Concepts; Testing Principles and Deployment; Computer based Systems; System Engineering Hierarchy – System Modelling and Simulation; Business Process Reengineering; Product Engineering; system modeling.
4. **Requirements Engineering and Modelling:** Requirements Engineering Tasks; Requirements Engineering Process; Eliciting Requirements; Developing Use-Cases; Analysis Modelling; Negotiating Requirements; and Validations. Requirements Analysis; Analysis Modelling Approaches; Object Oriented Analysis; Scenario Based and Flow Oriented Modelling.
5. **Design Concepts and Models:** Design concepts and principles**,** Software Design and Software Engineering, Design Context, Process and Quality; Design Concept – Abstraction, architecture, Pattern, modularity, information hiding, functional independence, refinement, design classes; design models – data elements, interface elements, architecture elements; User Interface Design- Process and Models, User Interface Design-The Golden Rules, Component-Level Design.

REFERENCES

Roger S. Pressman: Software Engineering – A practitioners' Approach. McGraw Hill
K.K. Aggarwal & Yogesh Singh: Software Engineering. New Age International Publishers
P. Jalote: Software Engineering. Narosa

CSC22: Object Oriented Programming

LEARNING OUTCOMES

Understand OOP concepts and features.
Make use of objects and classes for developing programs.
Learn to develop software using OO approach.

1. **OO Concepts:** Programming Paradigms: Unstructured Programming, Structured Programming, Object Oriented Programming; ADT; Class; Object; Message; Encapsulation; Polymorphism; Inheritance; Pros and Cons of Object-oriented Methodology; cin and cout Objects.
2. **Classes and Objects:** Classes; Friend Functions: Benefits and Restrictions, Friends Classes; Inline Functions; Constructor, Parameterized Constructor; Destructor and its usages; Static Data Member and Static Member Functions; Creating Object; Passing and Returning Object(s) to/from a Function; Object Assignment; Nested and Local Classes; Arrays of Objects; Pointer to Objects; this Pointer, Pointer to Derived Type; References; Reference vs Pointer; Reference Parameters; Dynamic Memory Allocation.
3. **Function and Operator overloading:** Function overloading: Rules, Overloading Constructors, Copy Constructors; Default Function Arguments vs. Function Overloading. Operator Overloading: Operators that cannot be Overloaded, Overloading Operators using Member Function and Friends Functions, Overloading different operators including prefix and postfix form of ++ and – operators.
4. **Inheritance & Virtual function**: Inheritance: Types of Inheritances, Base-Class Access Control, Protected Members, Protected Base-class Inheritance, Multiple Inheritance and problems, Solution to Multiple Inheritance Problem, Passing Parameters to Base Class Constructors; Virtual functions: Introduction, Calling a Virtual Function using Base Class Reference, Pure Virtual Function, Abstract Class.
5. **Generic Function, Exception and File Handling:** Generic Functions: Benefits, Functions with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a generic function, Restriction, Generic Classes. Exception Handling, user defined Exception. C++ Streams; C++ File Handling: Opening/Closing a File, Reading /Writing a Text File, Random Access, Reading /Writing Object to a File.

REFERENCES

Herbert Schildt: The Complete Reference C++. McGraw Hill
H.M. Deitel & P.J. Deitel: C++ How to Program. PHI
A. N. Kamthane: Object Oriented Programming with ANSI and TURBO C++. Pearson

CSC23: Advanced Data Structures

LEARNING OUTCOMES

Illustrate terminology and concepts of data structures.
Derive the mapping functions to maps the indices of multi dimensional arrays to index of 1D array.
Design the efficient algorithms for different matrix operations for various special matrices.
Design algorithm for various operations of different data structures.
Applications of various data structures to solve different problems.
Device applications based on Graph data structures.

1. **List and Matrices:** Data Structure, Linear Data Structure, Array Data Structure, Multi Dimensional Array, Mapping of Indices of 2D and 3D Arrays to the Index of 1D Array, Matrix, Mapping of Indices of Matrix Elements to One Dimensional (1D) Array Index, Special Matrices, Triangular, Diagonal, Tri-Diagonal, Representation in Row Major and Column Major Order, Mapping of non-null Elements in 1D Array, Sparse Matrix, Single Linked List, Circular Linked List, Doubly Linked List, Circular Doubly Linked List, Applications of Linked Lists: Bin Sort, Radix Sort, Convex Hull.
2. **Stacks and Queues:** Stack Data Structure, Various Stack Operations, Representation and Implementation of Stack using Array and Linked List, Applications of Stack: Conversion of Infix to Postfix Expressions, Parenthesis Matching, Towers of Hanoi, Rat in a Maze, Implementation of Recursive Functions, Queue Data Structure, Various Queue Operations, Circular Queue, Representation and implementation of queues using Array and Linked List, Applications of Queue Railroad Car Rearrangement, Machine Shop Simulation, Image-Component Labeling, Priority Queues: Priority Queue Using Heap; Max and Min Heap; Insertion into Heap; Deletion from a Heap; Applications of Priority Queue: Heap Sort.
3. **Trees:** Binary Trees and their Properties; Representation of Binary Trees: Array-Based and Linked Representations; Binary Tree Traversals; Binary Search Trees (BST); Operations on BST: Search, Insertion and Deletion; BST with Duplicates; Applications of BST: Histogramming, Best- Fit Bin Packing, AVL Trees; AVL Tree Representation; Introduction to Red-Black and Splay Trees, B-Trees and their Representation; Operations on B-Tree: Search, Insertion and Deletion; B+-Trees.
4. **Sorting, Searching, and Hashing:** Insertion Sort, Bubble Sorting, Quick Sort, Merge Sort, Shell sort, Sequential search, binary search, Introduction to Hashing, Hash Table Representation, Hash Functions, Collision and Overflows, Linear Probing, Random Probing, Double Hashing, and Open Hashing.
5. **Graphs and Disjoint Sets:** Graph Terminology & Representations, Graphs & Multi- graphs, Directed Graphs, Representations of Graphs, Weighted Graph Representations; Graph Traversal Methods: Breadth-First Search and Depth-First Search; Spanning Tree and Shortest Path Finding Problems, Disjoint Sets, Various Operations of Disjoint Sets, Disjoint Sets Implementation.

REFERENCES

Sartaj Sahni: Data Structures, Algorithms and Applications in C++. Universities Press
D. Samanta: Classic Data Structures, PHI
Narasimha Karumanchi: Data Structures and Algorithms Made Easy. CareerMonk

CSC24: Operating System and Shell Programming

LEARNING OUTCOMES

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.
Be familiar with various types of operating systems including UNIX, Linux and windows.

1. **Introduction:** Operating Systems functions, 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, Virtual Machines, Operating-System Generation, System Boot.
2. **Process, Threads & Scheduling:** Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, IPC Systems, Communication in Client-Server Systems, Threads: Overview, Multithreading Models, Thread Libraries, Threading Issues, Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Operating System Examples, Algorithm Evaluation.
3. **Synchronization & Deadlocks:** Background, The Critical-Section Problem, Peterson’s Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples, Atomic Transactions, Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.
4. **Memory Management Strategies:** Background, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory, Operating-System Examples.
5. **File-System & Shell Programming:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, NFS, The WAFL File System, Shell Programming: Types of shells, Shell functionality, Environment, writing & executing shell scripts, Debugging script, Making interactive scripts, Variables, default variables, Functions & file manipulations, Regular Expression & Filters.

REFERENCES

A. Silberschatz, P.B., Galvin, and G. Gagne: Operating System Concept. JW
W. Stallings: Operating Systems - Internals and Design Principles. Pearson

CSC26: Lab-III (OOP)

Implementation of at least ONE specific assignment concerning each of the following:

1. Structure and nested structure with methods.
2. Creation of class along with constructors, destructors.
3. Friend function, static data members and member functions
4. Passing objects to a function, returning object from a function and nested classes.
5. Creating array of objects, pointer to objects and pointer to class members.
6. Dynamic memory allocation and de-allocation using new and delete, this pointer & reference variable.
7. Function and constructor overloading, operator overloading, copy constructor, default function arguments.
8. Different types of inheritance, virtual functions, abstract class and exception handling.
9. Creation, use and deployment of generic functions
10. File creation and handling using FileStream classes, command line arguments

CSC27: Lab-IV(ADS+SP)

Implementation of at least ONE specific assignment concerning each of the following:

1. Creation of Matrix and Special Matrices classes OR Implementation of the different functions of the Matrix and Special Matrices.
2. Creation Array class OR implementation of the different functions of the Array D.S.
3. Creation of the Different types of Linked list classes OR Implementation of the various functions of the Different types of Linked list D.S.
4. Implementation of the bucket and radixSort algorithms using single linked list D.S.and convex hull using doubly circular linked list D.S.
5. Creation of the Generic Stack class and implementation of the infixTopostfix, EvaluatePostfix, tower of Hanoi problem, Rat on a maz problems, and recusive functions such as factorial(n), Fib(n), gcd(m, n), DecimalToBinary(n) using Stack D.S.
6. Creation of the Generic CircularQueue class and implementation of the Railroad Car Rearrangement, Image-Component Labeling using Queue.
7. Creation of Array and linked representation of the tree data structure class.
8. Creation of the linked representation of the Expression tree data structure.
9. Creation of the Heap tree data structure and implementation of the heapsort and priority Queue D.S.
10. Creation of the binary search tree data structure.
11. Implementation of various functions of the graph.
12. Implement the depth first search and breadth first search of a graph.
13. Implementation of the topological sort.
14. Use of Basic UNIX Shell Commands: ls, mkdir, cd, cat, touch, file, wc, sort, cut, grep etc.
15. Shell Programming based on control statements and operators.
16. Shell Programming for file handling.
17. Shell Programming for managing the access of the files and users.

CSC31: Data Communication and Computer Networks

LEARNING OUTCOMES

Identify the components required to build different types of networks.·
Choose the required functionality at each layer for a given application.
Identify solutions for each functionality at each layer.·
Trace the flow of information from one node to another node in the network.

1. **Data Communications & Physical Layer:** Data Communications, Data Representation, Data Flow, Networks & Types, Internet, Standards and Administration, Network Models, Protocol Layering, TCP/IP Protocol Suite, The OSI Model, Physical Layer: Data and Signals, Periodic Analog Signals, Digital Signals, Transmission Of Digital Signals, Transmission Impairment, Data Rate Limits, Noiseless Channel: Nyquist Bit Rate, Noisy Channel: Shannon Capacity, Performance: Bandwidth, Throughput, Latency (Delay), Bandwidth-Delay Product, Jitter, Bandwidth Utilization, Transmission Media, Guided Media, Unguided Media, Switching, Circuit-Switched Networks, Packet Switching.
2. **Data-Link Layer:** Nodes and Links, Services, Link-Layer Addressing, Address Resolution Protocol (ARP), Error Detection and Correction, Block Coding, Cyclic Codes, Cyclic Redundancy Check, Polynomials, Checksum, Forward Error Correction, Data Link Control (DLC), DLC Services, Data-Link Layer Protocols, HDLC, Point-To-Point Protocol (PPP).
3. **Media Access Control (MAC):** Random Access: Aloha, CSMA, CSMA/CD, CSMA/CA, Controlled Access, Channelization: FDMA, TDMA, CDMA, Wired LANS: Ethernet, Ethernet Protocol, Standard Ethernet, SONET, ATM, Wireless LANS, IEEE 802.11, Bluetooth, WiMAX, Connecting Devices, Virtual LANS.
4. **Network Layer:** Network-Layer Services, Packet Switching, Network-Layer Performance, IPv4 Addresses, Forwarding of IP Packets, Network-Layer Protocols: Internet Protocol (IP), ICMPv4, Mobile IP, Unicast Routing, Routing Algorithms, Unicast Routing Protocols, Multicast Routing, Next Generation IP, IPv6 Addressing, The Ipv6 Protocol, Transition from IPv4 To IPv6.
5. **Transport Layer & Application Layer:** Introduction, Transport-Layer Protocols, User Datagram Protocol, Transmission Control Protocol, SCTP, Application Layer: Introduction, Client-Server Programming, Standard Client-Server Protocols, World Wide Web and HTTP, FTP, Electronic Mail, Telnet, Domain Name System (DNS), SNMP.

REFERENCES

A. S. Tanenbaum: Computer Networks. PHI
William Stallings: Data and Computer Communications. Pearson
Behrouz A. Forouzan: Data Communications and Networking (SIE). TMH

CSC32: Artificial Intelligence

LEARNING OUTCOMES

Understand design of AI and Deep learning based systems and design AI based expert system.
Solve different AI puzzles such as Sudoku, 8-puzzle etc. in Prolog.
Build, train and apply fully connected deep neural networks.
To implement efficient recurrent neural networks for sequence based data.

1. **AI Techniques:** History, AI techniques, Problem solving using Search, Uninformed v/s Informed Search, Heuristic Search Techniques: Hill Climbing, Simulated Annealing, Best First Search: OR Graphs, Heuristic Functions, A* Algorithm, AND-OR Graphs, AO* Algorithm, Adversarial Search: Zero-sum perfect information Games, Optimal Decisions and Strategies in Games, Mini-max Algorithm, Alpha-beta Pruning, Solving game puzzles: NIM, Chess, Chinese Checkers, etc.
2. **Knowledge Representation & Reasoning:** Propositional logic, Inference in First order logic, Forward v/s Backward chaining, Resolution, Resolution-refutation; PROLOG: Logic Programming and Horn Clauses; CUT and FAIL operators, Built-in Goals, Negation, recursive Lists processing, PROLOG programs for puzzles; KR methods, Weak Slot-and-Filler Structures: Semantic Nets, Frames; Semantic Web and ontologies, Strong Slot-and-Filler Structures: Conceptual Dependency, Scripts.
3. **Connectionist Models:** Feed-forward, Feedback; Activation Functions; Loss functions; Different Models: McCulloch and Pitts Model; Feed-forward model: Perceptron Learning, Delta rule and error Backpropagation learning; Deep learning, CNN architecture, layers, and learning; Feedback architectures and learning: Hopfield Model and Memories, Hebbian Learning, Boltzmann Machines and Energy Computations, real-life applications.
4. **Recurrent Neural Network:** RNN architecture and learning; different models and their applications in sequence classification and text analytics, long short-term memory (LSTM), Gated recurrent Unit (GRU); Unsupervised learning with ANN: Competitive Learning, SOM: architecture, classifications, Implementation and training, Ensemble methods.
5. **Genetic Algorithm and Fuzzy Logic:** Encoding, fitness functions, genetic operators, reproduction, evolutionary strategies, Applications of GA; Fuzzy Logic: fuzzy sets: properties and operations; Fuzzy logic and fuzzy rules, Mamdani fuzzy rule inferencing mechanism, Fuzzy Systems, Neuro Fuzzy Systems, etc.

REFERENCES

Tom M. Mitchell: Machine Learning. McGraw Hill
E. Rich & K. Knight: Artificial Intelligence. TMH
Michael Nielsen: Neural Networks and Deep Learning (free online book)
Stuart Russel & Peter Norvig: Artificial Intelligence–A Modern Approach. Pearson
Padhy: Artificial intelligence and intelligent systems. Oxford University Press

CSC33: Information Security

LEARNING OUTCOMES

Articulate and perform principled security analysis, design, and implementations.
Elaborate and develop security policies and programs for security and BC-DR assurance.
Explain, differentiate, and evaluate different security architecture/models in place for security assurance.
Illustrate basic cryptographic techniques and their respective significance.
Devise customized programs for operations security and access control systems.

1. **Context, CBK, and Principles:** IT Security Importance and Opportunities; Multidisciplinary Approach; Contextualizing Information Security; IS Expertise & Business Systems. Security Management Practices: Security Architecture and Models; BCP; Law, Investigations, and Ethics; Physical Security; Operations Security; ACM Systems and Methodology; Cryptography; Telecommunications, Network and Internet Security; and Application Development Security; Twelve Security Principle and CIA triad.
2. **Security Management and BC-DRP: Security Policies:** Programme-Level, Programme-Framework, Issue-Specific and System-Specific Policies; Development and Management of Security Policies: Security Objectives, Operational Security and Policy Implementation; Policy Support Documents Regulations; Standards Taxonomy; Risk Analysis and Management; Responsible for Security? Business Continuity Plan; Disaster Recovery Planning: Identifying Recovery Strategies, Shared-Site Agreements, Alternate Sites, Additional Arrangements, Testing DRP.
3. **Security Architecture and Models: Defining TCB:** Rings of Trust; Protection Mechanisms in a TCB: System Security Assurance Concepts, Goals of Security Testing and Formal Security Testing Models; TCSE: Minimal, Discretionary, Mandatory and Verified Protection; Trusted Network Interpretation and TCSEC; Comparing ITSEC and TCSEC & ITSEC; CTCPEC, FCITS; CI Models: Bell-LaPadula Model, Biba Integrity Model and Advanced Models; PPO: SFR, EAL and The CEL**.**
4. **Cryptography:** Cryptography Needs and Significance, Terms and Concepts: Cyphertext, Cryptanalysis, Cryptosystem, Message Digest etc; STE and Substitution; Digesting Data; Digital Certificates, Certification and Envelop; Symmetric and Symmetric Cryptography; Root, Private and Public; Digital Cryptography – Hashing Functions, Block Ciphers and Implementation of PPK cryptography.
5. **Operations Security and Access Control Systems:** Operations Security Principles; Operations Security Process Controls; Operations Security Controls in Action; Information Owner, Discretionary Access Control, ACL, MAC, RAC; Principles of Authentication: The Problems with Passwords, Multifactor Authentication, Biometrics, Single Sign-On, Kerberos and Federated Identities; Remote User Access and Authentication.

REFERENCES

M. Merkow & J. Breithaupt: Information Security - Principles and Practices. Pearson
M. E. Whitman & H. J. Mattord: Principles of Information Security. CENGAGE
M. Palmer: Guide to Operating Systems Security. CENGAGE

CSE34: Analysis and Design of Algorithms

LEARNING OUTCOMES

Lean Asymptotic notation for representing the time complexity of an algorithm in order notations.
Learn various algorithm development approaches such as divide and Conquer, Dydnamic Programmin, Greedy etc.
To Device different algorithms using various approaches for well known problems.
Lean the concept of Intractable problems such as NP, NP Complete, NP Hard etc.

1. **Algorithms Analysis and Divide and Conquer Approach:** Time Complexity, Complexity Representation using Order Notations: Big-o (O), Theta (Θ), Big-Omega (Ω), Small-o (o) and Small-Omega (ω) Notations; Properties of Complexity Notations; Limit Approach to Determine Order, Master Theorem. Algorithm Design Techniques (ADT): Divide and Conquer Approach – Divide, Conquer, and Combine Steps; Design and Analysis of Binary Search (Recursive and Non-recursive), Merger Sort, Quicksort, and Strassen‘s Matrix Multiplication Algorithms.
2. **Dynamic Programming Approach:** Introduction to Dynamic Programming; Difference Between Divide-and- Conquer and Dynamic Programming Approaches; Binomial Coefficient Finding using Dynamic Programming; Dynamic Programming and Optimization Problems: Chained Matrix Multiplication and Longest Common Subsequence Problems; Travelling Salesman Problem.
3. **Greedy Approach:** Introduction to Greedy Approach; Components of Greedy Approach: Selection Procedure, Feasibility Check, and Solution Check; Minimum Spanning Tree Generation: Prim‘s and Kruskal‘s Algorithms; Dijkstra‘s Algorithm for Single-Source Shortest Paths; Scheduling: Single Server and Multi-Server Scheduling, Scheduling with Deadlines; Huffman Code; The Knapsack Problem (Greedy Approach vs Dynamic Programming): 0-1 Knapsack and Fractional Knapsack Problems.
4. **Backtracking Approach:** Introduction to Backtracking; Backtracking Technique: State Space Tree, Promising and Non-Promising Nodes, Pruned State Space Tree; Backtracking Algorithms for n-Queens, Sum-of-Subsets, Graph Coloring, and 0-1 Knapsack Problems.
5. **Branch-and-Bound Method and Intractable Problems:** Introduction to Branch-and-Bound Method; Solving 0-1 Knapsack Problem using Branch-and-Bound Method: Breadth-First Search with Branch-and- Bound Pruning, Best-First Search with Branch-and-Bound Pruning; Solving Traveling Salesman Problem using Branch-and-Bound Method. Intractable Problems: NP-hard and NP-complete problems, Some examples of NP hard and NP complete Randomized Algorithm with examples

REFERENCES

R. Neapolitan & K. Naimipour: Foundations of Algorithms. Jones & Bartlett
T. H. Cormen etc.: Introduction to Algorithms. PHI
E. Horowitz, S. Sahani, and S. Rajasekaran: Fundamentals of Computer Algorithms. Galgotia

CSC36: Lab-IV (AI)

Implementation of at least ONE specific assignment concerning each of the following:

1. Puzzle solving in PROLOG: 8-queens, Sudoku, 8-puzzle, etc.
2. Design rule based expert system for theorem proving.
3. Implement Fuzzy logic based expert system.
4. Implement Transfer Learning using ResNet50 and CIFAR-10.
5. Implement a hand written digit recognizer using FFNN for MNIST dataset.
6. Implement an image classifier using CNN for ImageNet dataset.
7. Implement sequence data classifier using a long short-term memory (LSTM) network for Japanese Vowels dataset.
8. Implement sequence to sequence model using GRU.
9. Implemeting a regression model.
10. Implement optimization functions using GA.

CSC37: Lab-VI(ADA)

Implementation of at least ONE specific assignment concerning each of the following:

1. Implementation of various divide and conquer algorithms.

2. Implementation of the dynamic programming based algorithms.

3. Implementation of the algorithms based on greedy approach.

4. Implementation of the algorithms based on backtracking approach.

5. Implementation of the algorithms based on branch and bound approach.

(Remark: Detailed syllabus of some Electives and CBCSE/CBCSS courses shall be designed and shared whenever floated for teaching – after BOS approval.)

Section-2.1: Elective Courses (Elective-I)

CSE25.1: Elective-I: Advanced DBMS

LEARNING OUTCOMES

Recap to use DBMS features and be familiar with advanced SQL usage
Understanding of Query Processing and Query Optimization
Be proficient with Transactions, Concurrency Control and Recovery Systems
Be exposed to parallel, distributed and deductive databases and object database systems

1. **Coping with System Failures:** Introduction to ADBMS, ACID properties, Issues and Models for Resilient Operation, Undo Logging, Redo Logging, Undo/Redo Logging, Logging Rules, Recovery using different Logging methods, Quiescent and NonquiescentCheck pointing a Log, Recovery with a checkpointed Log, Protecting against Media Failures, Nonquiescent Archiving, Recovery using an Archive and Log, Transactions in SQL,Serializability, Atomicity, Read-only Transactions, Dirty Reads, other Isolations Levels, Review of PL/SQL.
2. **Concurrency Control:** Serial and Serializable Schedules, Conflict-Serializability, Precedence Graphs and a Test for Conflict-Serializability, Enforcing Serializability by Locks, The Locking Scheduler, Two-Phase Locking (2PL), Locking Systems with several Lock Modes: shared and Exclusive Locks, Compatibility Matrices, Upgrading Locks, Update Locks, Increment Locks, An architecture for a Locking Scheduler, The Lock Table, Managing Hierarchies of Database Elements: Locks with Multiple Granularity, The Tree Protocol, Concurrency Control by Timestamps, Concurrency Control by Validation, Constraints and Triggers.
3. **Advanced Transaction Management:** Serializability and Recoverability, Recoverable Schedules, ACR, Logical Logging, Recovery from Logical Logs, View Serializability, Polygraphs and the Test for View-Serializability, Resolving Deadlocks, Deadlock Prevention by Ordering Elements and Timestamps, Distributed Databases: Distributed Commit, Two-phase Commit (2PC), Distributed Locking, Long-duration Transactions, Sagas and Compensating Transactions
4. **The Query Compiler:** Parsing, Estimating the cost of operations, Query optimization, Completing the Physical-Query-Plan and Query Execution; Storage management**.**
5. **Database System Architectures:** Object Definition Language (ODL), Object-relational Model,XML and its Data Model, Object-orientation in Query Languages,Logical Query Languages, Centralized and Client-Server Architectures,Parallel Databases,Spatial and Geographic Databases, Multimedia Databases, Mobility and Personal Databases.

REFERENCES

H. Garcia-Molina, J. D. Ullman, and J. Widom: Database Systems: The Complete Book. Pearson
A. Silberschatz, H. F. Korth, and S. Sudarshan: Database System Concepts. Mc Graw Hill
R. Ramakrishnan & J. Gehrke, Database Management Systems. Mc Graw Hill

CSE25.2: Elective-I: Computer Graphics

LEARNING OUTCOMES

Define the different concepts related to Computer Graphics Display Systems and devices.
Device algorithms for drawing lines, circles, and ellipse.
Learn various 2D, 3D transformation, clipping operations.
Lear various projection methods and curve and surface representation methods for representing the objects.

1. **Introduction:** Introduction of Computer Graphics and its Application; Overview of Computer Graphics; Video Display Devices; Raster Scan Display; Random Scan Display; Cathode Rays Tube (CRT) Display Device; Direct View Storage Tube (DVST) Display Device; Flat Panel display: Plasma Panel Display, Thin Film Electroluminescent Display, Light Emitted Diode (LED) Display Device, Liquid Crystal Display Device; Color CRT Display Devices.
2. **Algorithms:** Line Drawing Algorithm: Symmetric Digital Differential Analyzer (DDA), Simple DDA, Bresenham Line Drawing Algorithm; Circle Generating Algorithm, Mid-point Circle Algorithm; Mid-point Ellipse Algorithm; Polygon Fill Algorithm: Scan-Line Polygon Fill Algorithm, Boundary Fill Algorithm, Flood Fill Algorithm.
3. **Two Dimensional Geometric Transformation:** Basic Transformations: Translation, Rotation, Scaling; Matrix Representation; Homogeneous Coordinates; Composite Transformations: Translations, Scalings, General Pivot-Point Rotation, General Fixed-Point Scaling, General Composite Transformations and Computational Efficiency; Reflection, Reflection about x-axis, Reflection about y-axis, Reflection about a Line Perpendicular to xy-Plane and Passes Through Origin, Reflection about a General Line in xy-Plane, Shearing Operations; Transformation between Cartesian Coordinate Systems.
4. **Two Dimensional Viewing and Cliping:** Viewing, Window to Viewport Coordinate Transformation; Clipping: Point clipping; Line Clipping; Cohen-Sutherland Line Clipping Algorithm, Midpoint Subdivision Line Clipping Algorithm, Liang-Barsky Line Clipping Algorithm, Splitting Concave Polygon into Convex Polygons, Polygon Clipping: Sutheland-Hodgeman Polygon Clipping, Weiler-Atherton Polygon Clipping; Text clipping.
5. **Three Dimensional Geometric Transformations, Cliping, and Object Representations:** Translation, Rotation, Scaling, Reflections, Shears; Projection: Types of Projections, Perspective Projection, Orthographic Projection Transformation with Projection Plane as one of the Standard Plane, Orthographic Projection Transformation with Projection Plane Passes Through $R_0(x_0, y_0, z_0)$ and Normal Vector is $N=n_1i+n_2j+n_3k$, Isometric Projection Transformation; Three Dimensional Line Clipping; Curve Line and Surface: Spline Representation, Cubic Spline, Bezier Curve, Bezier Surface, Hidden Surface Detection: Z-Buffer Method, A-Buffer Method, Scan Line Method.

REFERENCES

Donald Hearn & M. Pauline Baker: Computer Graphics. Pearson
David F. Roger: Procedural Element for Computer Graphics. TMH
David F. Roger & J. A. Adams: Mathematical Element for Computer Graphics. TMH

CSE25.3: Elective-I: Theory of Computation

LEARNING OUTCOMES

Design automata and apply it for solving algorithmic problems.
Differentiate regular and non-regular languages, on relevant attributes.
Apply the concept of CFG, CFL and typical normal forms in solving problems.
Illustrate and design PDAs for given languages and vice-versa.
Design and test typical turing machines; and appreciate their practical significance.

1. **Finite Automata:** Concept of Automata, Computational Models and Formal systems; Automata Theory; Finite Automata: Deterministic Finite Automata (DFA), Languages of DFA; Non-Deterministic Finite Automata (NFA), Language of NFA, Equivalence of Deterministic and Non-deterministic Automata, Application of Automata: Finding String in Text, Recognizing a Set of Keywords, Finite Automata with Epsilon Transition.
2. **Regular Languages and Regular Grammars:** Regular Expressions, Finite Automata and Regular Expressions, Conversion from DFA to Regular Expression, Conversion from Regular Expression to Automata, Languages Associated with Regular Expressions, Connection between Regular Expressions and Regular languages, Regular Grammar, Properties of Regular Languages, Closure properties of Regular Languages. Identifying Non-regular Languages.
3. **Context Free Languages:** Context Free Grammars, Examples of Context Free Languages, Left most and Right most Derivations, Derivation Trees, Relationship between Derivation and Derivation Trees, Ambiguity in Grammars and Languages, Ambiguous Grammar, Methods for transforming Grammars; An useful Substitution Rule, Removing Useless productions, Removing λ- productions, Removing unit productions, Two important Normal Forms: Chomsky Normal Forms and Greibach Normal Form; Pumping Lemma for CFLs.
4. **Pushdown Automata:** Push Down Automata (PDA), Informal and Formal Definition of a Push Down Automata, Descriptions of a PDA, the Language Accepted by a Push Down Automata, Push Down Automata and Context Free Languages, Context Free Grammar for Push Down Automata; Deterministic Push Down Automata.
5. **Turing Machines:** The Standard Turing Machine, Definition of a Turing Machine, Turing Machine Language Accepters, Other Models of the Turing Machine, Multi-tape Turing Machines, Multidimensional Turing Machines, Nondeterministic Turing Machines, The Universal Turing Machine; Recursive and Recursively Enumerable languages, Some Problems that cannot be Solved by Turing Machines- Computability and Decidability.

REFERENCES

Peter Linz: An Introduction to Formal Languages and Automata. Narosa
J.E. Hopcroft, R. Motwani, and J. D. Ullman: Introduction to Automata Theory, Languages, and Computation. Pearson
J. C. Martin: Introduction to Languages and the Theory of Computation. TMH
K.L.P. Mishra and N. Chandrasekaran: Theory of Computer Science. PHI

CSE25.4: Elective-I: Software Quality Assurance

LEARNING OUTCOMES

Explain the context, concepts and typical SQAs.
Appreciate the role and perform configuration management activities.
Illustrate, use and hence evaluate different technical metrics of software.
Comprehend the role, and utility of formal methods in quality software development.
Explain and apply the cleanroom engineering during the quality assurance of software.

1. **Software Quality:** Quality Concepts, Quality Assurance, Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews , Formal Approaches to Software Quality Assurance, Statistical Software Quality Assurance, Software Reliability, Mistake-Proofing for Software, ISO 9000 Quality Standards, The ISO Approach to Quality Assurance Systems, and Software Quality Assurance Plan.
2. **Configuration Management:** Software Configuration: Baselines and Items; Software Configuration Management Process; Identification of Objects in the Software Configuration, Version Control, Change Control, Configuration Audit, Status Reporting, and Software Configuration Management Standards.
3. **Technical Metrics:** Software Quality, A Framework for Technical Software Metrics, Metrics for the Analysis Model, Metrics for the Design Model, Metrics for Source Code, Metrics for Testing, and Metrics for Maintenance.
4. **Formal Methods:** Basic Concepts, Deficiencies of Less Formal Approaches, Mathematical Preliminaries, Applying Mathematical Notation for Formal Specification, Formal Specification Languages, Using Z to Represent an Example Software Component and The Ten Commandments of Formal Methods.
5. **Cleanroom Engineering:** Cleanroom Approach, Cleanroom Strategy, Functional Specification, Cleanroom Design , Cleanroom Testing, and Certification.

REFERENCES

Roger S. Pressman: Software Engineering – A practitioners' Approach. McGraw Hill
K.K. Aggarwal & Yogesh Singh: Software Engineering. New Age International Publishers
P. Jalote: Software Engineering. Narosa

CSE25.6: Elective-I: Software Project Management

LEARNING OUTCOMES

Apply management skills and techniques to both commercial and government projects. .
Produce specific plans to manage the software development and maintenance efforts.
Evaluate software project management practices and recommend practical improvements.
Apply schedule and cost techniques to determine a Basis for estimate.
Understand the software process improvement, and quality management process.

1. **Introduction:** Introduction and Software Project Planning Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan, Software project estimation, Estimation methods, Estimation models, Decision process.
2. **Project Organization and Scheduling Project Elements**: Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques, Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts.
3. **Project Monitoring and Control:** Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Deskchecks, Walkthroughs, Code Reviews, Pair Programming.
4. **Software Quality Assurance:** Testing Objectives, Testing Principles, Test Plans, Test Cases, Types of Testing, Levels of Testing, Test Strategies, Program Correctness, Program Verification & validation,, Concept of Software Quality, Software Quality Attributes, Software Quality Metrics and Indicators, The SEI Capability Maturity Model CMM), SQA Activities, Formal SQA Approaches: Proof of correctness, Statistical quality assurance, Cleanroom process.
5. **Project Management and Project Management Tools:** Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, Change Control, Change Requests Management, Version Control, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, Cost Benefit Analysis, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

REFERENCES

Roger S. Pressman: Software Engineering – A practitioners' Approach. McGraw Hill
K.K. Aggarwal & Yogesh Singh: Software Engineering. New Age International Publishers
P. Jalote: Software Engineering. Narosa

CSE25.8: Elective-I: Data Mining and Warehousing

LEARNING OUTCOMES

Explain the context and concepts; and apply techniques related Data Mining and warehousing.
Learn various clustering and classification algorithms.
Able to identify the appropriate classification/ clustering techniques for a given dataset.
Becomes familiar with various data mining tools.

1. **Data Mining:** Introduction, Data warehouses, Transactional databases, Advanced Data Information Systems and Applications, Data Mining Functionalities, Classification of data mining systems, data mining task primitives, Integration of data mining systems with a data warehouse systems, Data Preprocessing: Descriptive data summarization, Data cleaning, Data Integration and Transformation, Data Reduction, Data discretization and Concept hierarchy generation.
2. **Data Warehouse and OLAP technology:** Multidimensional data model, Data Warehouse architecture and Implementation: OLAP, ROLAP, MOLAP, HOLAP etc., Data Cubes, Indexing OLAP data, OLAP queries, Discovery-driven exploration of data cubes.
3. **Frequent Patterns, Associations:** Association Rules, Frequent Itemsets, Closed Itemsets, Apriori algorithm, Generating association rules from frequent itemsets, Mining Closed Frequent Itemsets, Correlation Analysis, Metarule guided mining of Association Rules, Constraint Pushing, Classification v/s Prediction methods, Classification by Decision Tree Induction, Bagging and Boosting.
4. **Classification:** Bayesian classification, K- Nearest-Neighbor Classifier, Case-based Reasoning, Decision Tree, Random Forest, Backpropagation learning, Prediction: Linear v/s Non-linear Regression, Accuracy and Error measures: Hold-out method, Cross-validation, Bootstrap, estimating confidence intervals, Confusion matrix, ROC curves.
5. **Clustering:** Types of data in Cluster Analysis, Categorization of Clustering methods, Partitioning Methods: k-means, k-Medoids, CLARANS, Hierarchical Methods: BIRCH; Mining Time-series data, Introduction to Text Mining, Graph Mining, Social Network Analysis, and Web or Link Mining

REFERENCES

J. Han, M. Kamber, and J. Pei: Data Mining: Concepts and Techniques. Elsevier
I. Witten, E. Frank, and M. Hall: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier

CSE25.9: Elective-I: Software Testing and Verification

LEARNING OUTCOMES

Explain the context and concepts; and apply techniques related software testing.
Develop a viable and optimistic strategy for testing.
Perform object-oriented testing in practice.
Test web applications for its functionalities and environment.
Illustrate case tools and use them during testing and other phases.

1. **Testing Fundamentals:** Testing Terminology**,** Types of Testing; Basis Path Testing: Flow Graph Notation, Independent Program Paths, Deriving Test Cases and Graph Metrics; Control Structure Testing: Condition Testing, Data Flow Testing, and Loop Testing; Black Box Testing: Graph Based Testing methods, Equivalence Partitioning, Boundary Value Analysis; Objected Oriented Testing Methods: Testing Specialized Environments, Architecture and Applications: Testing GUI, Client Server Architecture, Documentation and Help Facilities, and Real Time Systems.
2. **Testing Strategies:** Strategic Approach to Software Testing: Verification and Validation, Organizing for Software Testing, Strategies for Conventional Architecture, Object Oriented Structure and Criterion for Completion of Testing; Strategic Issue; Significance and Potential; Testability; Unit and Integration Testing; OO Strategies: Unit Testing in OO Context, Integration Testing for OO Context; Validation Testing: Criteria, Configuration, Review, Alpha and Beta Testing; System Testing: Recovery Testing, Security Testing, Stress Testing, and Performance Testing; Art of Debugging: Debugging Process, Debugging Strategies and Fixing Errors.
3. **Object-Oriented Testing:** Broadening the View of Testing, Testing OOA and OOD Models, Object-Oriented Testing Strategies, Test Case Design for OO Software , Testing Methods Applicable at the Class Level , Interclass Test Case Design: Multiple Class and Tests Derived from Behavior Model.
4. **Testing Web Applications:** Testing Concepts for Web Applications; Dimensions of Quality, Errors in Web App Environment, Testing Strategy and Test Planning; Testing Process Overview; Content Testing: Objectives and Database Testing; User Interface Testing: Testing Strategy, Interface Mechanism, Interface Semantics, Usability Testing and Compatibility Tests; Component Level Testing:; Navigation Testing: Syntax and Semantics; Configuration Testing: Server and Client Side Issues; Security Testing; Performance Testing: Objectives, Load and Stress Testing.
5. **CASE:** Concept of CASE, Building Blocks for CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The Integration Architecture, The CASE Repository, The Role of the Repository in I-CASE, Features and Content**.**

REFERENCES

Roger S. Pressman: Software Engineering – A practitioners' Approach. McGraw Hill
K.K. Aggarwal & Yogesh Singh: Software Engineering. New Age International Publishers
P. Jalote: Software Engineering. Narosa

Section-2.2: Elective Courses (Elective-II)

CSE35.1: Elective-II: Distributed Systems

LEARNING OUTCOMES

Understand the characteristics, issues and importance of distributed systems.
Understand the architecture and processes of Distributed System.
Design a distributed system that fulfills requirements with regards to key distributed systems properties.
Understand the importance of security in distributed systems.
Understand and analyze synchronization processes.

1. **Concept of Distributed Systems:** Goals – Connecting users with resources, Transparency, Scalability, Openness; Hardware Concepts: Multiprocessors, Multicomputer, Homogenous and Heterogeneous, Multicomputer; Software Concepts: Distributed Applications, Distributed OS, NOS and Middleware; Client Software model; Clients and Servers, Application Layering, Client-Server Architectures.
2. **Architecture Styles:** Centralized Architectures, Decentralized Architectures and Hybrid Architectures; Architecture vs Middleware – Interceptors and General Approaches to Adaptive Software; Self Management in Distributed Systems - Feedback Control Model; Systems Monitoring with Astrolabe, Differentiating Replication and Globule; Automatic Component Repair Management in Jade.
3. **Processes:** Threads, Threads in Distributed Systems; Virtualization, Architectures of Virtual Machines and Clients; Networked User Interfaces - Client-Side Software for Distribution Transparency; Servers - General Design Issues, Server Clusters and Managing Server Clusters; Code Migration - Approaches to Code Migration, Migration and Local Resources and Migration in Heterogeneous Systems.
4. **Communication:** Fundamentals - Layered Protocols and Types of CommuUication; RPC - Basic RPC Operatio, Parameter Passing, Asynchronous RPC and Example of DCE RPC; Message Oriented Communication - Transient Communication and Persistent Communication; Stream Oriented Communication - Support for Continuous Media, Streams and Quality of Service, Stream Synchronization; Multicast Communication - Application-Level Multicasting and Gossip-Based Data Dissemination.
5. **Naming and Synchronization Clock:** Names, Identifiers and Addresses; Flat Naming, Home-Based Approaches, Distributed Hash Tables and Hierarchical Approaches; Structured Naming, Name Resolution, Name Space and Domain Name System; Attribute-Based Naming, Hierarchical Implementations LDAP and Decentralized Implementations. Synchronization Clock Physical Clocks, Global Positioning System and Clock Synchronization Algorithms; Logical Clocks - Lamport’s Logical Clocks, Vector Clocks; Mutual exclusion, Centralised, Decentralised, Distributed and Token Ring; Global Positioning Nodes; Election Algorithms.

REFERENCES

A. S. Tennenbaum: Distributed Operating Systems. Pearson
G. Coulouris, J. Dollimore, and T. Kindberg: Distributed Systems: Concepts and Design. Pearson
J. M. Crichlow: Distributed Systems - Computing Over Networks. PHI

CSE35.2: Elective-II: Cryptography

LEARNING OUTCOMES

Understanding of the cryptography and security trends.
Understanding of different services provided by the various cryptographic techniques
Symmetrical and Asymmetrical cryptography, their applications and standards,
Understanding of the Data integrity, Authentication, and Digital Signatures.

1. **Introduction to Cryptography**: Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Recommended Reading and Web Sites. Symmetric Ciphers-Classical Encryption Techniques, Substitution , techniques, Transposition Techniques, Rotor Machines, Steganography.
2. **Block Ciphers and the Data Encryption Standard :** Block Cipher Principles, The Data Encryption Standard, The Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles. Finite Fields- Groups, Rings, and Fields, Modular Arithmetic, The Euclidean Algorithm. Advanced Encryption Standard - Evaluation Criteria for AES. The AES Cipher, Multiple Encryption and Triple DES, Block Cipher Modes, Stream Ciphers and RC4.
3. **Symmetric and public key Encryption :** Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation, Public-Key Encryption and Hash Functions. Introduction to Number Theory- Prime Numbers, Fermat's and Euler's Theorems
4. **Public-Key Cryptography and RSA**: Principles of Public-Key Cryptosystems, The RSA Algorithm Chapter ,Digital Signatures and Authentication Protocols - Digital Signatures, Authentication Protocols, Digital Signature Standard .
5. **Key Management**: Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic, Elliptic Curve Cryptography. Message Authentication and Hash Functions- Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and MACs.

REFERENCES

William Stallings: Cryptography and Network Security: Principals and Practice. Pearson
Johannes A. Buchmann: Introduction to Cryptography. Springer
Bruce Schiener: Applied Cryptography. Wiley

CSE35.4: Elective-II: System Software

LEARNING OUTCOMES

Understand machine structure IBM 360/370 and others.
To be able to code using IBM 370 machine language.
To understand the design of Assembler.
To understand the design of Macro Processor.
To understand the design of Loader.
To understand the design of Editors, Debuggers, and Profiler.

2. **Machine Structure:** Introduction to Machine Structure, General Approach to a New Machine; Structure of IBM 360/370 Machines: Memory, Registers, Data, Instructions, and Special Features; VAX Architecture; Pentium Pro Architecture; UltraSPARC Architecture; PowerPC Architecture; Cray T3E Architecture.
3. **Machine Language:** Introduction to Machine Language, A Sample Program to Add a Number to Contents of 10 Adjacent Full-words in the Memory without Looping, Same Program by Address Modification using Instruction as Data, Same program by Address Modification using Index Register, Same program by Looping and Index Register; Assembly Language: Introduction to Assembly Language, Benefits of Assembly Language over Machine Language, Pseudo-op, USING vs. BALR Instructions, An Assembly Language Program, Program Using Literals.
4. **Assembler:** Introduction to Assembler, Element of Assembly Language Programming, Assembly Language Statements, Pass Structure of Assembler; Design of One pass Assembler for IBM PC; Design of Two Pass Assembler for IBM 360 Machine: Statement of Problem, Data Structure, Format of Data Structure, Algorithms.
5. **Macro Processor:** Macro Instruction, Macro Instruction Arguments, Conditional Macro Expansion, Macro Calls within Macros, Macro Instructions Defining Macros; Implementation of two Pass Macro Processor; Implementation of Single Pass Macro Processor; Implementation of Macro Calls Within Macros.
6. **Loader:** Introduction to Loader; Loading Scheme: Compile and Go Loader, General Loading Scheme, Absolute Loader, Subroutine Linkage, Relocating Loader, Direct Linking Loader, Binders, Dynamic Loading, Overlay, Dynamic Linking; Design of Absolute Loader; Design of Direct Linking Loader: Problem Specification, Specification of Data Structure, Format of Data Bases, Algorithm. Software Tools: Software Tools for Program Developments, Program Design and Coding, Program Entry and Editing, Program Testing and Debugging; Design of Software Tools; Profilers, Editors; Screen Editors; Word Processors; Structure Editors; Design of Editors; Debug Monitors; Testing Assertions.

REFERENCES

John J. Donovan: Systems Programming. TMH
Leland L. Beck & D. Manjula: System Software - An Introduction to Systems Programming. Pearson
D. M. Dhamdhere: System Programming and Operating Systems. TMH

CSE35.5: Elective-II: Digital Image Processing

LEARNING OUTCOMES

Understand the need for image transforms different types of image transforms and their properties.
learn different techniques employed for the enhancement of images.
learn different causes for image degradation and various image restoration techniques.
Understand the need for image compression and to learn spatial and frequency domain techniques of image compression.
Learn different Image Segmentation techniques.
Learn Image classification using Deep learning.

1. **Introduction and Spatial filtering:** Components of a DIP system, Elements of visual perception, Light and Electromagnetic Spectrum, Image sensing, acquisition, Sampling and Quantization, Spatial and Intensity resolution, Basic relationships between pixels, Basic mathematical tools used in DIP, Intensity transformation functions, Histogram processing, Lowpass and Highpass Spatial Filters, 2D and 3D Convolutions
2. **Frequency domain filtering and Image Restoration:** 1-D and 2-D Discrete Fourier Transform (DFT) and IDFT, Lowpass, Highpass, selective Frequency Domain filters, Fast Fourier Transform (FFT), Model of Image restoration and Reconstruction, Noise models, Restoration in presence of noise only-Spatial filtering, Max, Min, Mean, Median filters, Periodic Noise reduction using frequency domain filtering, Notch filter.
3. **Color Image Processing:** Color Models: RGB, HSI, CMYK, YCbCr, Full Color Image Processing, Color image smoothing, sharpening, Using color in image segmentation, Intro to wavelet and other image transform, Introduction to Morphological Image Processing
4. **Image Compression:** Relative Data Redundancy (RDR), Compression Ratio, Coding Redundancy, Fidelity criteria, Lossless v/s Lossy Image Compression, Image compression Model, Huffman coding, Run-length coding, Bit plane coding, Image formats and standards, KLT, JPEG-steps etc., , Watermarking
5. **Image Segmentation and Classification:** Point, Line, and Edge detection, thresholding, Edge and Region based segmentation, Region Segmentation using Clustering, Feature extraction: Boundary and Region feature descriptors, Principal components, whole image features etc, Image datasets, Image Classifiers using Neural Networks, Deep learning, Deep Convolutional Neural Networks etc.

REFERENCES

R. C. Gonzalez and R. E. Woods: Digital Image Processing. Pearson
R. C. Gonzalez, R. E. Woods, and S. Eddins: Digital Image Processing using MATLAB. Mc Graw Hill
M. Sonka, V. Hlavac, and R. Boyle: Image Processing, Analysis, and Machine Vision. Vikas Publishing House
Ashwin Pajankar: Python 3 Image Processing. Bpb

CSE35.7: Elective-II: Cyber Security

LEARNING OUTCOMES

Explain and encapsulate the context of cyberspace, cybercrime and cybersecurity in the current landscape.
Perform risk assessment and devise management program.
Illustrate different access and authentication techniques.
Compare and design technical security management program.
Perform threat and incident assessment along with contextual management.

1. **Cyber Security Practices and Standards:** Cybercrime and Cyberoffences;Cyberspace and Cybersecurity, Standards and Best Practices, Good Practice for Information Security, ISO/IEC 27000 Suite of Information Security Standards, ISO 27001and ISO 27002; NIST Cybersecurity Framework and Security; CIS Critical Security Controls for Effective Cyber Defense; COBIT 5 for Information Security; Payment Card Industry Data Security Standard (PCI DSS); ITU-T Security Documents; Effective Cybersecurity: Management Process, Best Practices and Standards Documents;
2. **Risk Management:** Risk Assessment Concepts, Challenges, and Management; Asset Identification; Threat Identification: STRIDE Threat Model, Threat Types, and Sources; Control Identification; Vulnerability Identification, Categories, National Vulnerability Database and Common Vulnerability Scoring System; Risk Assessment Approaches; Likelihood Assessment; Impact Assessment; Risk Determination; Risk Evaluation; Risk Treatment: Reduction, Retention, Avoidance, and Transfer; Risk Assessment Best Practices
3. **System Access:** System Access Concepts and Authorization; User Authentication: Models, Means and Multifactor Authentication; Password-Based Authentication; Possession-Based Authentication; Biometric Authentication; Risk Assessment for User Authentication; Access Control : Subjects, Objects, and Access Rights, Policies, Discretionary, Role-Based and Attribute-Based, Metrics; Customer Access; System Access Best Practices.
4. **Technical Security Management:** Security Architecture; Malware Protection Activities; Malware Protection Software; Identity and Access Management; Intrusion Detection; Data Loss Prevention; Digital Rights Management; Cryptographic Solutions; Cryptographic Key Management; Public Key Infrastructure; and Technical Security Management Best Practices.
5. **Threat and Incident Management:** Technical Vulnerability Management; Security Event Logging; Security Event Management; Threat Intelligence; Cyber Attack Protection; Security Incident Management Framework; Security Incident Management Process; Emergency Fixes; Forensic Investigations; Threat and Incident Management Best Practices.

REFERENCES

William Stallings: Effective Cybersecurity - A Guide to Using Best Practices and Standards. Pearson
N. Godbole & S. Belapure: Cyber Security - Understanding Cyber Crimes, Computer Forensics and Legal Perspectives. Wiley
R. Jones: Internet Forensics: using Digital Evidence to Solve Computer Crime. O’Reilly Media
C. Steel: Windows Forensics: The Field Guide for Conducting Corporate Computer Investigations. Wiley

CSE35.8: Elective-II: Cloud Computing

LEARNING OUTCOMES

Understand the concept of cloud computing.
Understand the various types, technologies and standard involve in cloud computing.
Able to setup cloud computing environment.
Understand the applications of the cloud computing technology.

1. **Cloud Computing Basics:** Introduction to Cloud Computing; Concept of Utility Computing; Feature, Attributes, Characteristics of Cloud Computing; Benefits and Limitations of Cloud Computing; Components of Cloud Computing; Organizational Scenarios of Cloud Administering and Monitoring Cloud Services.
2. **Computing Architecture:** Cloud Computing Architecture; Delivery Models; Deployment Models; cloud service model; SOA and the Cloud; Virtualization; Designing Web Applications with Cloud Support; Amazon Web Services.
3. **Virtualization:** Introduction, Characteristics of virtualized environments: Increased security,Managed execution,Portability,Taxonomy of virtualization techniques: Execution virtualization,Other types of virtualization,Virtualization and cloud computing, Pros and cons of virtualization: Advantages of virtualization,The other side of the coin: disadvantages, Technology examples: Xen: paravirtualization, VMware: full virtualization,Microsoft Hyper-V.
4. **Cloud Applications and Cost Model:** Scientific applications: Healthcare: ECG analysis in the cloud, Biology: protein structure prediction, CRM and ERP, Productivity, Social networking, Pricing model in Cloud Computing.
5. **Advanced Topics in Cloud Computing:** Energy efficiency in clouds: Energy-efficient and green cloud computing architecture, Market-based management of clouds: Market-oriented cloud computing, A reference model for MOCC, Technologies and initiatives supporting MOCC, Federated clouds/Inter-Cloud, Characterization and definition, Cloud federation stack, Aspects of interest, Technologies for cloud federations.

REFERENCES

R. Buyya, C. Vecchiola, S.T. Selvi: Mastering Cloud Computing: Foundations and Applications Programming. Elsevier
J. Hurwitzet al.: Big Data for Dummies. Wiley
Tom White: Hadoop: The Definitive Guide. O'Reilly
E. Capriolo, D. Wampler, and J. Rutherglen: Programming Hive, Data Warehouse and Query Language for Hadoop. O'Reilly
Alan Gates: Programming Pig. O'Reilly

CSE35.9: Elective-II: Machine Learning

LEARNING OUTCOMES

Understand the concept of machine leaning.

Understand the various types, technologies and standard involve in machine learning.

Able to use different machine learning techniques for classification and regression problems.

Understand the applications of the various machine learning techniques.

1. **Machine Learning and applications:** Intelligent Systems, Types of Learning: Supervised, Unsupervised, and Reinforcement Learning, Applications of ML: social network analysis, web mining, natural language processing, online fraud detection, speech recognition, product recommendations, malware filtering, etc.
2. **Supervised Learning:** Supervised Machine Learning Algorithms: Kernelized Support Vector Machines, Linear and Non-linear SVM; Linear and Non-linear Regression, Multiple Regression, Log-linear regression models, Regression trees, Random forest.
3. **Unsupervised Learning**: Types of Unsupervised Learning, Challenges in Unsupervised Learning, Partitioning algorithms: Hierarchical algorithms: Agglomerative, Divisive methods; Density/grid based methods: DBSCAN, Probabilistic and generative models: Mixture of Gaussians; Expectation Maximization (EM) Clustering, Measuring quality of clustering.
4. **Genetic Algorithm and Applications:** evolutionary strategies, differential evolution, co-evolution, multi-objective GA (MOGA), Neuro-Genetic hybrid algorithm; Swarm Intelligence: Introduction, Swarm Based versus Population based techniques, Particle Swarm Optimization, Ant Colony Optimization.
5. **Machine Learning with python:** Colab Notebook, NumPy, SciPy, matplotlib, pandas, Keras, Classifying Iris dataset, Model Evaluation and Improvement: Cross-Validation, Cross-Validation in scikit-learn, Benefits of Cross-Validation, Stratified k-Fold Cross-Validation and Other Strategies, Grid Search, Overfitting the Parameters and the Validation Set, Grid Search with Cross-Validation, Evaluation Metrics and Scoring: Metrics for Binary Classification, and Multiclass Classification, Regression Metrics.

REFERENCES

Tom M. Mitchell: Machine Learning. McGraw Hill

A. C. Müller & S. Guido: Introduction to Machine Learning with Python. O'Reilly

S. Rakasekharan & G. A. Vijayalakshmi: Neural Networks, Fuzzy Logic and Genetic Algorithms. PHI

S. Russel & P. Norvig: Artificial Intelligence–A Modern Approach. Pearson

CSE35.14: Elective-II: Ad-Hoc Networks

LEARNING OUTCOMES

Understand the concept of ad-hoc and sensor networks, their applications and network architectures.

Understand the various types, technologies and standard involve in Ad-hoc networks.

Understand the Principles and QoS in routing of mobile Ad-Hoc Networks with security constraints.

Set up and evaluate measurements of protocol performance in different types of networks.

1. **Adhoc Wireless Networks:** Cellular and Ad Hoc Wireless Networks , Applications of Ad Hoc Wireless Networks, Computing Emergency Operations Wireless Mesh Networks, Wireless Sensor Networks, Hybrid Wireless Networks, Issues in Ad Hoc Network, Medium Access Scheme, Routing, Multicasting , Transport Layer Protocols, Quality of Services, Addressing and Service Discovery, Energy Management, Scalability, Ad Hoc Wireless internet.
2. **Body, Personal and Local Ad hoc Wireless Networks**: Mobile Ad-hoc –Body Area Network, Personal Area Network and Wireless Local Area Network; Technologies for Ad-hoc networks, IEEE 802.11 Architecture and Protocols – EEE 802.11 DCF, EEE 802.11 RTS/CTS; IEEE 802.16 Standard, A technology for WBAN and WPAN –Bluetooth, A Bluetooth Network, Bluetooth data transmission.
3. **Multicasting Techniques in Mobile Ad-hoc Networks:** Multicast protocols in wired Networks-Shortest path Multicast Tree, Core based trees Multi Cast Protocols, Multicast Protocols in Mobile Ad-hoc Networks- On-Demand Multicast Routing Protocol(ODMRP), Multicast Ad-hoc on-demand Distance Vector Routing Protocol (Multicast AODV), Forwarding Group Multicast Protocol(FGMP) , Core-Assisted Mesh Protocol.
4. **Quality of Service in Mobile Ad-Hoc Networks** : Operating Principles, Routing in Mobile Ad-hoc Networks, Routing with Quality of service Constraints, QoS Routing in Ad-hoc Networks, QoS Routing with Security Constraints .
5. **Cross Layer Design and Integration -**Cross layer Design: Need for cross layer design, cross layer optimization, parameter optimization techniques, Cross layer cautionary perspective, Co-operative networks:- Architecture, methods of co-operation, co-operative antennas, Integration of ad hoc network with other wired and wireless networks.

REFERENCES

Mohammad Ilyas: The Handbook of Ad Hoc Wireless Networks. CRC press

C. S. R. Murthy & B. S. Manoj: Ad Hoc Wireless Networks: Architectures and Protocols. Pearson

C. E. Perkins: Ad hoc Networking. Addison-Wesley

I. S. S. Basagni, M. Conti, and S. Giordano: Mobile Ad Hoc Networking. Wiley

CSE35.15: Elective-II: Programming with Java

LEARNING OUTCOMES

Understand the concept and syntax of Java Programming.
Understand the various features of Java Programming language.
Able to understand various concepts related to Java.
Able to write Java programs for stand alone as well as applet programs.
Able to connectivity of various DBMS with Java applications.

1. **Introduction, Environment and Programming Structure:** Java White Paper Buzzwords, History of Java, Common Misconceptions, Choosing a Development Environment: Command-Line Tools, Running a Graphical Application, Building and Running Applets; A Simple Java Program, Comments, Data Types, Variables, Operators, Input and Output, Control Flow, Big Numbers, Arrays.
2. **Class, Objects and Inheritance:** Introduction to OOP, Predefined Classes, User Defined Classes, Static Fields and Methods, Method Parameters, Object Construction, Packages, Class Path, Documentation Comments, Class Design; Inheritance: Super-classes and Subclasses, Types of Inheritance, Polymorphism, Abstract class, Object: The Cosmic Super class, Generic Array Lists, Object Wrappers and Autoboxing, Methods with a Variable Number of Parameters, Enumeration Classes, Reflection, Inheritance Guidelines, Interfaces.
3. **String Handling, Exception Handling and Generic Programming:** String Handling APIs: String, Immutable String, Methods of String Class, StringBuffer, StringBuilder, StringTokenizer. Exceptions: Dealing with Errors, Catching Exceptions, Guidelines for Using Exceptions, Assertions, Logging; Generic Programming: Definition, Generic Methods, Bounds for Type Variables, Generic Code and VM, Restrictions and Limitations, Inheritance Rules for Generic Types, Reflection and Generics.
4. **Java Collections and Multithreading:** Collection Interfaces, Concrete Collections, The Collections Framework, Algorithms, Legacy Collections, Multithreading: Threads, Interrupting Threads, Thread States, Thread Properties, Synchronization, Blocking Queues, Thread-Safe Collections, Callable and Futures, Executors, Synchronizers.
5. **Java GUI Programming and JDBC:** Introduction to Swing, Creating a Frame, Positioning a Frame, Displaying Information in a Component, Displaying Images, Event Handling, Basics of Event Handling, Actions, Mouse Events, The AWT Event Hierarchy; JDBC: Basic JDBC Programming Concepts, JDBC Drivers, Statements, Executing Queries, Result Sets.

REFERENCES

Horstmann & Cornell: Core Java Volume I: Fundamentals. Pearson
Horstmann & Cornell: Core Java Volume II: Advanced Features. Pearson
H. Schildt: Java 2: The Complete Reference. TMH
Dietel & Dietel: Java How To Program. Pearson
Bruce Eckel: Thinking in Java. Pearson
Balagurusamy: Programming with Java: A Primer. TMH

Section-3.1: CBCS Courses (Sem-I)

CBCSE17.1: Statistical Computing

LEARNING OUTCOMES

To apply discrete and continuous probability distributions to various business problems.
To perform Test of Hypothesis as well as calculate confidence interval and understand the concept of p-values.
To learn non-parametric test such as the Chi-Square test for Independence as well as Goodness of Fit.
To compute Bivariate and Multivariate Regression and Correlation and perform ANOVA and F-test.

1. **Overview of R:** R data types and objects, reading and writing data**;** Control structures, functions, scoping rules, dates and times**;** Loop functions, debugging tools**;** Simulation, code profiling.
2. **Solution of Equations and System of Simultaneous Equations:** Solution of Algebraic and Transcendental Equations using Bisection, Regula False, and Newton Raphson Methods, Gauss Elimination, Gauss Seidel, and Jacobi Methods.
3. **Interpolation, Numerical Differentiation and Integration, and Differential Equations:** Interpolation using Lagrange, and Newton's methods, Extrapolation, Least Square Fitting, Numerical Integration using Trapezoidal, and Simpson's Rules, Numerical Solution of Ordinary Differential Equations using Euler's and Range-Kutta Methods.
4. **Statistics:** Population, Sample, Sample Collection Methods, Data Representations and Classification, Central Tendency and Dispersion: Mean, Geometric Mean, Harmonic Mean, Median and Mode, Quartiles and Percentiles, Measures of Dispersion: Range, Variance, Standard Deviation, and Coefficient of Variation.
5. **Probability and Hypothesis Testing:** Sample Space, Events, Equally Likely Events, Probability, Independent Events, Addition and Multiplication Rules, Conditional Probability, Probability Distributions – Normal, Binomial, and Poison Distributions; Correlation using Karl Pearson and Spearman Method; Hypothesis Testing: t-Test, Chi-Square Test, Analysis of Variance (ANOVA), F-Test.

REFERENCES

S.C. Chapra & R.P.Canale: Numerical Methods for Engineering. TMH
V. Rajaraman: Computer oriented numerical methods. PHI
Andy Field, J. Miles, and Z. Field: Discovering Statistics Using R. SAGE
A. S. Grewal: Higher Engineering Mathematics. Khanna
M. K. Jain, S. R. K. Iyengav, and R. K. Jain: Numerical Methods for Scientific and Engineering Computation. New Age

CBCSE17.2: e-Business Systems

LEARNING OUTCOMES

Explain Systems; and IT related terminology, trends, challenges, obstacles and prospects.
Illustrate and perform work-centered analysis on business systems.
Devise factoring algorithms, analyze and develop their improved versions.
Implement and analyze different array-based searching and sorting algorithms.
Design well-styled programs, trace and test the same.

1. **Systems and IT:** Systems, Business System, Phases in Building and Maintaining Systems, IT-Based Innovations in Every Business Function, Product Design Systems, CAD Software, Procurement Systems, Supply Chain Management, Electronic Data Interchange, Manufacturing , Sales and Marketing Systems, Delivery Systems, Customer Service Systems, Finance Systems, Dramatic Progress in Processing Data, Recent Trends in IT; Applying IT to the Real World.
2. **Business Systems:** Frameworks and Models, Viewing Businesses as Systems, Businesses as Systems Consisting of Business Processes, The Value Chain , Business Processes and Functional Areas of Business , Information Systems and Work Systems, Increasing Overlap Between Information Systems and Work Systems, Framework for Thinking About Any System in Business , Work-Centered Analysis Framework, Five Perspectives for Viewing a Work System , Architecture: System Components and How They Operate Together, Performance: How Well the System Operates, Analyzing an IT-Enabled System From a Business Professional's Viewpoint, Work-Centered Analysis Method, Limitations and Pitfalls
3. **Business Processes and Models**: Business Processes, Process Modeling: DFD, Flowcharts and Structured English, Process Characteristics: Degree of Structure, Range of Involvement, Level of Integration, Rhythm, Complexity, Degree of Reliance on Machines etc, Communication and Decision Making; Evaluating Business Process Performance: Activity Rate and Output, Consistency, Productivity, Cycle Time, Downtime and Security, Basic Communication and Decision Making Concepts.
4. **Typical Information Systems:** Information System Categories related to Specific Functional Areas of Business, IS Categories applicable Functional Areas; Office Automation Systems; Communication Systems: Teleconferencing, E-Mail, Fax, SMS, Groupware, Internet, Intranets, Extranets, Knowledge Management, and Group Support Systems, Transaction Processing Systems, MIS and Executive Information Systems, Decision Support Systems, Case Based Reasoning, and Intelligent Systems, Enterprise Systems, Limitation and Uses of Typical Information Systems.
5. **Customer, Product and IT:** Customer's View of Product and services, The Customers' Experience, Evaluating Products and Services: Cost, quality, responsiveness, Reliability and Conformance to standards, Performance Variables of IT: Functional Capabilities and Limitations, Ease of use, Compatibility and Maintainability, Approaches of Organizational Computing-Centralized, Personal, Distributed, Networked and Client-Server, Current Limits of Software; Types of Software, Programming viewed as Business Process, Major Developments in Programming – Special purpose, Spreadsheets and CASE, Artificial Intelligence and Intelligent Systems.

REFERENCES

Steven Alter: Information Systems – The Foundations of E-Business. Pearson
S. Haag & M. Cummings: Information Systems Essentials. Mcgraw-Hill
R. C. Nickeson: Business Information Systems. Prentice Hall

CBCSE17.3: Human Computer Interaction

LEARNING OUTCOMES

Acquire first-hand experience of useful HCI techniques in practice.
Comprehend the latest topics in multimedia, global information systems, and the web-based models for rich interaction.
Articulate social and contextual models and theories related to HCI design processes,
Appreciate and illustrate the role of interaction design, universal access, and rich interaction.
Apply design principles and established rules during user-friendly interface designs.

1. **HCI Overview:** Need, Issues in Human Computer Interaction and Significance; Overview of Human Sensory Capabilities and Limitations: Input-Output Channels and Design Focus; Analysis of Design Experiments; Human Sensory Capabilities and Limitations: Memory, Types of Memory, Comparative association with Computer memory, Thinking and Emotions; Human Sensory Capabilities and Limitations: Individual Differences and Psychology of Design of Interactive Systems.
2. **Computers**: Text Entry, Design Focus- Numeric Pads, Positioning, Pointing and Display Devices; Display Devices, Devices for Virtual Reality and 3D interactions, Physical Controls, Sensors and Special Devices; Paper: Printing and Scanning, Readability of Text, Font Issues etc.
3. **Interaction:** Models of Interaction, Framework for HCI and Ergonomics; Industrial Interfaces; Interaction Styles: Introduction and Types with Examples, Navigation, Elements of WIMP Interface, Interactivity and its context, Experience and edutainment etc; Interaction Paradigms: Introduction, Different Paradigms of Interaction, Timesharing, Batch Processing, Personal computing, Distributed etc; Language vs Action; Hypertext, Multimodality, WWW, Sensor based Context-aware Interactions etc.
4. **Interaction Design Basics:** Introduction, Design Basics and Process of Design, Scenarios, Navigation Design: Local structure, Global Structure, Dialog etc; Interaction Design Basics: Introduction, User Focus, Cultural Probes and Scenarios; Screen Design, Alignment and Layout, Screen Colors etc.
5. **Interaction Design Rules**: Introduction, Principles to Support Usability: Learn ability, Flexibility, and Robustness; Design Rules: Standards, need, significance; Underlying Theory, Usability factors etc.; Design Guidelines: Fundamental Guidelines, Golden Rules and Heuristics; Sheiderman Eight Rules; Norman’s seven Principles for Transformation; HCI Patterns

REFERENCES

Alan Dix et al.: Human-Computer Interaction. Pearson
Carlo et al.: Human Computer Interaction. PHI
B. J. Mosley & B. A. Posey: Just Enough Software Test Automation. Prentice Hall

Section-3.2: CBCS Courses (Sem-II)

CBCSE28.1: Modeling and Simulation

LEARNING OUTCOMES

Define the basics of simulation modeling and replicating the practical situations in organizations
Generate random numbers and random variates using different techniques.
Develop simulation model using heuristic methods.
Analysis of Simulation models using input analyzer, and output analyzer
Explain Verification and Validation of simulation model.

1. **Basics:** Concepts of Systems, Models, and Simulation. Distributed Lag Model, Cobweb Models; The process of a simulation Study, Exponential Growth Models, Exponential Decay Models, Type of simulation, Discrete-Event Simulation: Time-Advance Mechanisms, Components and Organization of a Discrete Event Simulation Model.
2. **Simulation Problems**: Monte Carlo Method, **Discrete** Simulation and Continuous Simulation and their Examples; **Discrete Simulation**: Simulation of Inventory problem, Simulation of Single-Server Queuing System, **Continuous Simulation:** Pure-pursuit Problem.
3. **Random Number Generators:** Linear Congruential Generators, Other kinds of Generators, Testing Random-Number Generators; Generating Random Variates: Various Approach Approaches.
4. **Output Data Analysis for a Single System & Simulation Languages:** Transient and Steady-State Behavior of a Stochastic Process, Type of Simulations with regard to output Analysis and Statistical Analysis for Testing Simulation; Comparisons of Simulation Packages with programming languages Introduction to different types of Simulation Languages. Factors in Selection of discrete system simulation; Object-Oriented Simulation.
5. **Verification and Validation:** Model Building, Verification of Simulation Models: Validating first-time model, Subsystem validity, internal validity, sensitive analysis, face validity; Calibration and Validation of Models, Validation of Model Assumptions, Validating Input, Output Transformations.

REFERENCES

Geoffrey Gordon: System Simulation. PHI
M. Law & W. D. Kelton: Simulation Modeling and Analysis. Mc Graw Hill
Fred Maryanski: Digital Computer Simulation. Hayden Book Co
Jerry Banks: Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice. Wiley
P. B. Zeigler: Theory of Modelling and Simulation. Krieger
J. Banks et al: Discrete Event System Simulation. Pearson

CBCSE28.2: Business Informatics

LEARNING OUTCOMES

Explain the concepts related to business analytics and its utility in decision making.
Comprehend database analytics to enrich business analytics processes.
Perform descriptive analytics on business data in context of practices.
Comprehend and use ways of business forecasting.
Apply data mining techniques to solve pertinent business problems and decisions.

1. **Business Analytics:** Informatics and Business Analytics, Using Business Analytics, Impacts and Challenges, Evolution of Business Analytics, Analytic Foundations, Modern Business Analytics, Software Support and Spreadsheet Technology, Descriptive, Predictive, and Prescriptive Analytics, Data for Business Analytics; Big Data, Data Reliability and Validity; Models in Business Analytics: Descriptive, Predictive and Prescriptive; Model Assumptions, Uncertainty and Risk, Problem Solving with Analytics: Interpreting Results and Making a Decision, Implementing the Solution.
2. **Database Analytics:** Data Sets and Databases, Using Range Names in Databases; Data Queries: Tables, Sorting, and Filtering; Database Functions: Logical Functions, Lookup Functions for Database Queries, Template Design, Data Validation Tools, Form Controls, PivotTables, PivotTable Customization and Slicers.
3. **Descriptive Analytics:** Data Visualization, Value of Data Visualization, Tools and Software for Data Visualization, Creating Charts , Charts from PivotTables, Geographic Data, Data Visualization Tools; Descriptive Statistics: Metrics and Data Classification, Frequency Distributions; Percentiles and Quartiles, Cross-Tabulations, Descriptive Statistical Measures, in Business Decisions; Measures of Dispersion; Chebyshev's Theorem and the Empirical Rules; Measures of Association; Using Descriptive Statistics to Analyze Survey Data; Statistical Thinking in Business Decisions and Variability in Samples.
4. **Business Forecasting:** Qualitative and Judgmental Forecasting, Historical Analogy, The Delphi Method, Indicators and Indexes, Statistical Forecasting Models, Forecasting Models for Stationary Time Series, Moving Average Models, Error Metrics and Forecast Accuracy, Exponential Smoothing Models, Forecasting Models for Time Series with a Linear Trend, Double Exponential Smoothing, Regression-Based Forecasting for Time Series with a Linear Trend, Forecasting Time Series with Seasonality and Regression-Based Seasonal Forecasting Models.
5. **Business Data Mining:** Data Mining, Cluster Analysis, Measuring Distance Between Objects, Normalizing Distance Measures, Clustering Methods, Classification, An Intuitive Explanation of Classification, Measuring Classification Performance, Classification Techniques, k-NN, Discriminant Analysis, Association and Cause-and-Effect Modeling.

REFERENCES

J. R. Evans: Business Analytics. Pearson
D. Delen & E. Turban: Business Intelligence, Analytics, and Data Science: A Managerial Perspective. Pearson
Steven Alter: Information Systems: Foundations of e-Business. Pearson

CBCSE28.3: Social Informatics

LEARNING OUTCOMES

understand knowledge of issues and problems in social informatics.

issues of impacting on uptake of digital systems and innovations by diagnosing problems in relations between technologies and use in a range of application domains.

Apply appropriate principles and methodologies to address challenges in the design and deployment of novel digital systems.

different disciplinary perspectives on social informatics and ability to apply them to solve design and deployment challenges.

Devise plan and execute requirements investigations and system evaluation studies from a social informatics perspective, and present findings in a clear and effective manner.

1. **Introduction to Social Informatics:** Defining Social Informatics, Value of Social Informatics, Disconnection Between Popular and Scholarly Discourse, Fundamental Ideas of Social Informatics- Theoretical Approaches, Direct Effects Theories, Predictions of Paperless Offices, Varied Effects of ICT Use.
2. **The Field of Social Informatics:** The Socio-technical Character of ICTs-Designing and Configuring Systems, The designing of ICTs continues during their use, Social design of ICTs, Approaches to Designing ICTs for Workplaces. The Consequences of ICTs for Organizations and Social Life, Communicative and computational roles of computer systems, There are important temporal and spatial dimensions of ICT consequences, ICTs are interpreted and used in different ways by different people, ICTs enable and constrain social actions and social relationships, ICTs and the control of users, ICTs important political consequences, negative consequences of ICT developments for some stakeholders.
3. **ICTs rarely cause social transformations:** Background and Case Studies, ICT Policy Analysis, Major ICT Public Policy Analysis Organizations or Programs, The Computer Science and Telecommunications Board, Information Technology Advisory Committee, ICT Research Institute', Programme on Information and Communication Technologies (PICT), Commission's Information Society Project**,** ICT Policy Analysis in the Next Decades, Public Access to the Next Generation Internet (NGI): A Social Informatics View.
4. **Teaching Key Ideas of Social Informatics** : Why teach Social Informatics, Social informatics teaching in the context of broad trends in science-oriented education, Summarizing the Teaching of Social Informatics, Current Status of Teaching Social Informatics, Issues with the Current Status of Teaching Social Informatics, Key Social Informatics Issues, Social Informatics as Informed Critical Thinking, Issues with teaching social informatics, needs of the many ICT-oriented disciplines
5. **Communicating Social Informatics Research to Professional and Research Communities : Purpose,** Audience, Communicating to ICT professional audiences, Perceptions of the relevance of social informatics research, Competition for the attention of the ICT professional audience, Strategies for communicating to ICT professional audiences, Communicating to Academic and Research Communities, Challenges of communicating to academic and research communities, Strategies for improving communication with other academic and research communities.

REFERENCES

R. Kling et al.: Learning from Social Informatics: Information and Communication Technologies in Human Contexts. Center for Social Informatics, Indiana University

Pnina Fichman & Howard Rosenbaum: Social Informatics: Past, Present and Future. Cambridge Scholars Publishing

Section-3.3: CBCS Courses (Sem-III)

CBCSE38.3: Multimedia Applications

LEARNING OUTCOMES

1. **Multimedia Primer:** Basic Concepts, Multimedia Storage Devices, Multimedia Highway, Multimedia Applications; Stages in Multimedia Development; Multimedia Development Requirements, Multimedia Skills Development and Expertise Requirements.
2. **Multimedia Text, Sound, Images, and Video:** Text-Power and Meaning, Fonts and Faces, Using Text in Multimedia; Computers and Text, Font Editing and Design Tools, Hypermedia and Hypertext; Sound-Power of Sound, Multimedia Systems Sound, Digital Audio, Making MIDI Audio, Audio File Formats, MIDI vs Digital Audio, Sound in Multimedia Applications, Music CDs and Audio Production Guidelines. Still Images and Vector Graphic, Bitmaps, Vector Drawings, 3-D Drawing and Rendering, Color and Image File Formats; Animation-Power, Principles, Techniques, File Formats, Developing Animation; Video-Using Video, Working of Videos, Analog Standards, NTSC, PAL, SECAM and ATSC DTV; Digital Display Standards; Digital Video, Video recording etc; Shooting, Editing Video; Storyboarding, Platform, Lighting, Chroma Keys etc; and Optimizing Video File Storages.
3. **Multimedia Hardware and Software:** Multimedia Platforms, Connections- SCSI,IDE,USB and Firewire; Multimedia Storage Devices, MM Input-Output Devices, Multimedia Communication Devices; Multimedia Software Tools – Text, Editing and WP tools, OCR Software, Drawing and Painting Tools, 3-D Modeling and Animation Tools; Image and Sound Editing Tools; Animation, Video and Digital Movie Tools; and Multimedia Accessories.
4. **Multimedia Authoring and Integration:** Developing Multimedia Applications, Types of Authoring Systems: Object Based, Icon Based, Page based, Card-based, Stages of Authoring, Editing, Organizing, Interactivity, Performance Tuning, Cross Platform Features, Cross Platform Authoring Notes; Authoring Notes; Introduction to MX Flash MX / Director MX.
5. **Multimedia for WWW:** Internet, Internetworking, Bandwidth Issue, Internet Services, WWW and HTML, Web Pages, Dynamic Webpages and XML; Multimedia Web; Web Servers, Browsers and Search Engines; Web Page Makers and Site Builders, Plug-in and Delivery Vehicles; Beyond HTML, 3D Worlds; Multimedia on the Web- Workspace, Nibbling, HTML and multimedia; Text for Web, Images for the web – GIF, PNG Images, JPEG, Image Maps; Sound and Animation on the Web. Miscellaneous topics and supplements.

REFERENCES

Vaughon: Multimedia – Making it Work. TMH
Parekh: Principles of Multimedia. TMH
Li and Drew: Fundamentals of Multimedia. Pearson

Section-4.1: Ability/ Skill Enhancement courses (Sem-II)

CBCSS29.1: Programming with Python

LEARNING OUTCOMES

Understand the basic construct of Python programming language
Apply various constructs and control structures in problem solving
Understand the object-oriented program design and development in Python
Write clear and effective python code
Access database using python programming

6. **Introduction:** Getting Started: Setting up Programming Environment, Python on Different Operating Systems, Running Python Programs from a Terminal. Variables & Simple Data Types: Variables, Strings, Numbers, Comments, The Zen of Python. Working with Lists: What is a List, Changing, Adding, Removing Elements, Organizing a List, Avoiding Index Errors, Looping through an Entire List, Avoiding Indentation Errors, Making Numerical Lists, Slicing a List, Working with Tuples and Dictionaries.
7. **Basic Constructs:** User Inputs: input() and int() Functions, Accepting Input in Python. Conditional Tests: if Statements, Using if Statement with Lists. While Loop: Introducing while Loops, using a flat, break, continue, Using a while Loop with Lists and Dictionaries.
8. **Functions, Classes, & Modules:** Functions: Defining a Function, Passing Arguments, Return Values, Passing a List, Passing an Arbitrary Number, Storing Your Functions in Modules. Classes: Creating and Using a Class, Working with Classes and Instances, Inheritance, _init_() Method for a Child Class, Overriding Methods, Instances as Attributes, Importing Classes, Modules, Storing Multiple Classes in a Module, Importing Classes from a Module, Importing a Module into a Module.
9. **Files & Exceptions:** Reading from a File, Reading an Entire File, File Paths, Reading Line-by-line, Making a List of Lines from a File, Working with a File's Contents, Large Files, Writing to a File, Writing to an Empty File, Writing Multiple Lines, Appending to a File. Exceptions: Handling the ZeroDivisionError Exception, Using try-except Blocks, Using Exceptions to Prevent Crashes, The else Block, Handling the FileNotFoundError Exception, Analyzing Text, Working with Multiple Files, Failing Silently, Deciding with Errors to Report, Storing Data, Using json.dump() and json.load(), Saving the Reading User-Generated Data, Refactoring.
10. **Advanced Python Features:**Data Visualization: Generating Data, Installing matplotlib, Plotting a Simple Line Graph, Rolling Dice with Pygal, Making a Histogram. CSV File Format: Parsing CSV File Headers, Extracting and Reading Data, Plotting Data in a Temperature Chart, the datetime Module, Plotting Dates. Working with APIs: Using a Web API, Git and GitHub, Requesting Data using an API Call, Installing Requests, Using GIT for Version Control. Overview of Django.

REFERENCES

Eric Matthes: **Python Crash Course: A Hands-On, Project-Based Introduction to Programming**. No Starch Press
Mark Lutz: Learning Python. O'Reilly
Zed A. Shaw: Learn Python the Hard Way. Addison-Wesley

CBCSS29.2: Mobile Applications

Learn to set up a new Material App using Android Studio.
Understand the Widget tree and learn to use pre-made Flutter, widgets for user interface design.
Learn to incorporate Image and Text Widgets to create simple user interfaces.
Learn to Customise pre-built Flutter widgets.
Adding App Icons for iOS and Android builds.
Learn to run Flutter apps on iOS Simulator, Android Emulator and physical iOS and Android devices.

1. **Introduction:** Frameworks and Tools for Mobile App Development, Characteristics of Mobile Applications, History of Mobile Application Frameworks and Tools, Introduction to Android, iOS, and Flutter. Client-Server Architecture:1-, 2-, 3-tier, types of Connection, Synchronization, Mobile Device Types, Mobile Device Components, Types of Mobile Applications
2. **Mobile Application Development using Flutter**: to set up a new Material App using Android Studio, Creating UI with Flutter: Using Hot Reload and Hot Restart to quickly refresh the app UI and understand when to use each, using the Pubspec.yaml file to incorporate, dependencies, custom assets and fonts, an introduction to the Widget build() method, using layout widgets such as Columns, Rows, Containers and Cards, incorporating Material icons using the Icons class
3. **Building Apps with State:** Understanding the difference between Stateful and Stateless widgets and when they should each be used, understanding how callbacks can be used to detect user interaction in button widgets, declarative style of UI programming and how Flutter widgets react to state changes, importing dart libraries to incorporate additional functionality, variables, data types and functions work in Dart, building flexible layouts using the Flutter Expanded widget, relationship between setState(), State objects and Stateful Widgets.
4. **Using the Dart package manager:** to use Dart package manager to incorporate Flutter compatible packages into your projects, functions in Dart and the arrow syntax, to refactor widgets and understand Flutter's philosophy of UI as code. Structuring Flutter Apps: to use Dart Constructors to create customisable Flutter widgets, apply common mobile design patterns to structure Flutter apps.
5. **Security:** User to Mobile Client Security Issues, Mobile Client Security Issues, Client-Server Communications Security Issues, Existing Web Architectures and Back-End Systems Security Issues, Mobile Application Development Management.

REFERENCES

V. Lee, H. Schneider, and R. Schell: Mobile Applications: Architecture, Design, and Development. Pearson
Marco L. Napoli: Beginning Flutter: A Hands On Guide to App Development. Wiley
Bill Phillips & Brian Hardy: Android Programming the Big Nerd Ranch Guide. Big Nerd Ranch
Brian Fling: Mobile Design and Development. O'Reilly

Section-4.2: Ability/ Skill Enhancement courses (Sem-III)

CBCSE39.1: Data Analytics and Hadoop

LEARNING OUTCOMES

Understand the various parts of Hadoop condition, for instance, Hadoop 2.7, Impala, Yarn, MapReduce, Pig, Hive, HBase, Sqoop, Flume, and Apache Spark

Learn Hadoop Distributed File System (HDFS) and YARN building, and make sense of how to function with them for limit and resource organization

Understand MapReduce and its qualities and retain advanced MapReduce thoughts

Ingest data using Sqoop and Flume

Get a working learning of Pig and its parts

1. **Cloud Computing Basics:** Introduction to Cloud Computing; Concept of Utility Computing; Feature, Attributes, Characteristics of Cloud Computing; Benefits and Limitations of Cloud Computing; Components of Cloud Computing; Organizational Scenarios of Cloud Administering and Monitoring Cloud Services
2. **Computing Architecture:** Cloud Computing Architecture; Delivery Models; Deployment Models; cloud service model; SOA and the Cloud; Virtualization; Designing Web Applications with Cloud Support; Amazon Web Services
3. **Big Data and Hadoop:** What is Big Data; Limitations of existing architectures; How Hadoop solves these problems; Hadoop Ecosystem; Core components of Hadoop; Hadoop functions: Perform, Read and Write; Rack Awareness
4. **Hadoop Architecture and HDFS:** Hadoop 2.x Cluster Architecture – Federation and High Availability; Resource Management in Hadoop; Hadoop Configuration Files; Hadoop Cluster Modes; Password-less SSH; Data Loading Techniques and Analysis; Basic HDFS Commands; Running a MapReduce Job on Hadoop
5. **MapReduce:** Use Cases of MapReduce; Difference between MapReduce and Traditional Techniques; Hadoop 2.x architecture and components; Execution flow of YARN MapReduce application; Input Splits concept; MapReduce Job Submission Flow; Combiner and Partitioners; Counters in MapReduce; Map and Reduce Side Joins; MR Testing Framework; Distributed Cache Concept; Custom Input Formatting MapReduce; Sequence Input Format in MapReduce, Pig: Use cases for using Pig; Conceptual data flow; Pig Execution; Data Models; Querying with Pig Operators.

REFERENCES

R. Buyya, C. Vecchiola, and S. T. Selvi: Mastering Cloud Computing: Foundations and Applications Programming. Elsevier

J. Hurwitzet al.: Big Data for Dummies. Wiley

Tom White: Hadoop: The Definitive Guide. O'Reilly

E. Capriolo, D. Wampler, and J. Rutherglen: Programming Hive, Data Warehouse and Query Language for Hadoop. O'Reilly

Alan Gates: Programming Pig. O'Reilly

CBCSE39.2: MATLAB Computations

LEARNING OUTCOMES

Understanding the features of the MATLAB development environment

Use the MATLAB GUI effectively

Design simple algorithms to solve problems

Write simple programs in MATLAB to solve scientific and mathematical problems

Use MATLAB effectively to analyze and visualize data.

Apply a top-down, modular, and systematic approach to design, write, test, and debug

Demonstrate understanding and use of fundamental data structures.

Create and control simple plot and user-interface graphics objects in MATLAB.

1. **Vectors and Matrices:** MATLAB Desktop Environment, Data types, Variables and Assignment Statements, Numerical Expressions, Operator precedence, Random number generation, Characters and Encoding, Relational Expressions, Creating matrix variables, dimensions, Scalar and Array Operations on Vectors and Matrices, Matrix Multiplication, Logical vectors, Meshgrid functions, Saving workspace, Importing and Exporting data
2. **MATLAB Scripts:** Scripts with Input and Output, Scripts to Produce and Customize simple plots, Introduction to File Input/Output (Load and Save), User-Defined Functions that Return a Single Value, Commands and Functions, Vectors and Matrices as function arguments, Selection statements: if, else, elseif, switch, Loop statements: while, for, Vectorizing Code, MATLAB Program Organization, Application: Menu-Driven Modular Program, "is" Functions in MATLAB Variable, Scope, M-files etc.
3. **String Manipulation and Cell Arrays:** Creating String Variables, Operations on Strings, "is" Functions for Strings, Converting between String and Number Types, creating Cell Arrays, Referring to and Displaying Cell Array Elements and Attributes, Storing Strings in Cell Arrays, Structures and operations,
4. **Advanced File Input and Output:** Opening and Closing a File, file identifier and file modes, Reading from file: fscanf, fgets, fgetl, textscan, Lower-Level File I/O Functions, Writing and Reading Spreadsheet Files, Using MAT-files for Variables
5. **Advanced problem solving with MATLAB:** 2-D plot types, logarithmic scale plots, pie charts, and histograms, customizing plots using cell arrays and string functions, 3-D plot functions, Built-in statistical and set operations, Sorting and Indexing, Sights and Sounds, programming GUIs etc.

REFERENCES

Stormy Attaway: MATLAB, A Practical Introduction to Programming and Problem Solving. Elsevier

Amos Gilat: MATLAB: An Introduction with Applications. Wiley

Cleve Moler: Numerical Computing with MATLAB. SIAM

CBCSE39.3: Web Based Programming

LEARNING OUTCOMES

Understand core concepts of J2EE programming.
Ability to J2EE concepts to real-world enterprise application development
Learn the concepts of Servlets, Java Server Pages, Database Connectivity, Enterprise Java Beans, and JavaMail APIs.

1. **J2EE Overview & Multi-tier Architecture:** Overview of J2SE, J2EE, Advantages of Java, Birth of J2EE, Why J2EE; Distributed Systems, The Tier, J2EE Multi-tier architecture, Implementation of Client-tier, Web-tier, EJB-tier, and EIS-tier, Challenges; J2EE best practices: Enterprise Application Strategy, The Enterprise Application - Client, Session Management, Web-tier and JSPs, EJB-tier, MVC, The Myth of Using Inheritance, Maintainable Classes, Performance Enhancement, Power of Interfaces, Threads, and Notification
2. **Java Servlets & JDBC:** Overview of HTML, XML, and XHTML, Java and XML, Parsing XML, Java Servlets and CGI Programming, A Simple Java Servlet, Anatomy of a Java Servlet, Life Cycle of the Servlet, Deployment Descriptor, Reading data from client, reading HTTP request headers, working with cookies, Tracking sessions. Overview of JDBC, JDBC Drivers, JDBC Packages, JDBC Process, Database Connection, Statement, ResultSet, Transaction Processing, Servlet program with JDBC.
3. **Java Server Pages:** Overview of JSP, JSP versus Servlet, JSP Tags: Variables and Objects, Directives, Scripting Elements, Standard Actions, Implicit Objects, Scope, Java Server Pages with Beans, Tomcat, User Sessions, Cookies, Session Objects, JSP with JDBC, Creating Custom JSP Tag Libraries.
4. **Enterprise Java Beans:** The EJB Container, EJB Classes, EJB Interfaces, Deployment Descriptions: Anatomy, Environment elements, referencing EJB, Sharing resources, Security elements, Query elements, Relationship elements, Assembly elements. Session Java Beans - stateless vs stateful, Entity Java Beans - Container-managed persistence, Bean-managed persistence. Message-driven Beans, JAR, WAR, EAR Files.
5. **JavaMail, CORBA and RMI:**JavaMail API and Java Activation Framework, Protocols, Exceptions, Send Email Message, Retrieving Email Messages, Deleting Email Message. CORBA : The Concept of Object Request Brokerage, Java IDL and CORBA, The IDL Interface. Java RMI: Remote Method Invocation Concept, Server Side, Client Side.

REFERENCES

Jim Keogh: J2EE : The Complete Reference. Mc Graw Hill
H. Schildt: Java 2: The Complete Reference. Mc Graw Hill
Kogent Solutions Inc.: Java Server Programming Java EE 7 (J2EE 1.7), Black Book, Dreamtech Press
Subrahmanyam Allaramaju et al.: Professional JSP J2EE 1.3 Edition. Wrox Press
K. Qian et al.: Java Web Development Illuminated. Narosa
Robert W. Sebesta: Programming the World Wide Web. Pearson