Learning Outcomes based Curriculum Framework (LOCF)

M.Sc.

in

COMPUTER SCIENCE

as a Subject

2022

PANSKURA BANAMALI COLLEGE

(Autonomous)

Choice Based Credit System (CBCS)

LEARNING OUTCOME-BASED CURRICULUM FRAMEWORK (LOCF)

Contents

1. Program Outcomes

These outcomes describe what students are expected to know and be able to do by the time of graduation. They relate to the skills, knowledge, and behaviors that students acquire in their course through the program. The Master of Science in Computer Science program enables students to attain, by the time of graduation:

- ✓ Demonstrate the aptitude of Computer Programming and Computer based problem solving skills.
- ✓ Display the knowledge of appropriate theory, practices and tools for the specification, design, implementation
- ✓ Ability to learn and acquire knowledge through online courses available at different MOOC Providers.
- ✓ Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study.
- ✓ Display ethical code of conduct in usage of Internet and Cyber systems.
- ✓ Ability to pursue higher studies of specialization and to take up technical employment.
- ✓ Ability to formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate .
- ✓ Ability to operate, manage, deploy, configure computer network, hardware, software operation of an organization.
- ✓ Ability to present result using different presentation tools.
- ✓ Ability to appreciate emerging technologies and tools.
- ✓ Apply standard Software Engineering practices and strategies in real -time software project development
- ✓ Design and develop computer programs/computer -based systems in the areas related to algorithms, networking, web design, cloud computing, IoT and data analytics.
- ✓ Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems Page .
- ✓ The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.
- ✓ The ability to work independently on a substantial software project and as an effective team member.

Curriculum
for
Master of Computer Science (MSC)
of
Panskura Banamali College (Autonomous).

Affiliated under Vidyasagar University.

FIRST YEAR FIRST SEMESTER

2. Course Outcomes

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
COS-101	Data Structure and Algorithm	3	1	0	40	10	4
COS-102	Advanced Computer Architecture	3	1	0	40	10	4
COS-103	Data Communication and Computer Network	3	1	0	40	10	4
COS-104	Software Engineering and Project Management	3	1	0	40	10	4
COS-191	Data Structure Lab	0	0	6	40	10	3
COS-192	Network Programming Lab	0	0	6	40	10	3
		12	4	12	240	60	22

Total Period/Week = 28 Total Marks = 300

FIRST YEAR SECOND SEMESTER

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
COS-201	Advanced Database Management System	3	1	0	40	10	4
COS-202	M-I: Automata Theory M-II: Compiler Construction	2 2	0 0	0 0	20	05 05	4
COS-203	M-I:OOPS with Java M-II: Programming in R	2 2	0 0	0 0	20 20	05 05	4
COS-204 (CBCS)	M-I: Computer Fundamentals M-II: Programming Concepts	2 2	0 0	0 0	20 20	05 05	4
COS-291	DBMS Lab	0	0	6	40	10	3
COS-292	M-I:OOPs Lab M-II: R lab	0 0	0 0	3 3	20 20	05 05	3
		14	2	12	240	60	22

Total Period/Week = 28 Total Marks = 300

SECOND YEAR FIRST SEMESTER

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
COS-301	Advanced Operating System	3	1	0	40	10	4
COS-302	M-I: Computer Graphics M-II: Image Processing	2 2	0 0	0 0	20 20	05 05	4
COS-303	Elective-I	3	1	0	40	10	4
COS-304 (CBCS)	M-I: DBMS M-II: Internet Technology	2 2	0 0	0 0	20 20	05 05	4
COS-391	M-I: Graphics Lab M-II: OS Lab	0 0	0 0	3 3	20 20	05 05	3
COS-392	M-I: Industrial Tour / Interview Skills M-II: Term Paper	0 0	0 0	0 3	0 0	25 25	3
		14	2	9	200	100	22

Total Marks = 300

Total Period/Week = 25

SECOND YEAR SECOND SEMESTER

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
COS-401	Artificial Intelligence	3	1	0	40	10	4
COS-402	Elective-II	3	1		40	10	4
COS-403	Elective-III	3	1	0	40	10	4
COS-491	AI Lab	0	0	3	20	05	2
COS-492	Project Work	0	0	10	0	100	6
COS-493	Grand Viva	0	0	0	0	25	2
		9	3	13	140	160	22

Total Period/Week = 25 Total Marks = 300

List of electives:

Elective 1, Code: COS-303
(A) Graph Theory
(B) Distributed computing
(C) Mobile Computing
(D) Pattern Recognition
(E) Machine Learning
(F) Soft Computing
(G) Recent Computing Architecture Trends

Elective II, Code : COS-402
(A) Web Technology
(B) Data Mining
(C) Parallel Computing
(D) Cryptography and Stenography
(E) Bio Informatics
(F) Natural Language Processing
(G) Data Science

Elective III, Code : COS-403
(A) Information Security
(B) Cognitive Computing
(C) Embedded System
(D) Multimedia
(E) Computational Geometry
(F) Combinatorial Algorithm
(G) Cloud Computing

COS-101: Data Structure and algorithm

Course Outcomes :

On successful completion of this course, students will be able to

- recall the fundamentals of data structure with their implementation and its applications.
- infer the complexity of algorithms.
- apply the Greedy methods to solve real time problems.
- compare different sorting and searching techniques.
- design new algorithms with Dynamic Programming Techniques for Analytical Problems.

Fundamentals of Linear and Non-Linear Data Structures .
Basic concepts about Algorithms, Data Structures, Recursion, Iteration, Big-O Notation, Brief

Foundations and Applications of Stacks, Queues, Arrays, Linked Lists – Singly, Doubly, and Circular

Linked Lists, Trees – Definitions, Representations, Binary Tree and Its Usefulness, Binary Search

Tree, Tree Traversal, Threaded Binary Trees, Binary Tree Representation of any Tree other than

Binary Tree, Decision Trees, Balanced Tree Schemes – AVL Trees, 2-3 Trees.

Searching- Basic concepts about Searching, B-Trees, Hashing.
Sorting- Different Sorting Algorithms and their complexity issues.
Advanced Data Structures- Binomial Heaps, Fibonacci Heaps, Amortized Analysis of Algorithms,
Disjoint Set Maintenance Techniques.

References:

1. Fundamentals of Data Structures in C by Horowitz, Sahni & Anderson-Freed, 2e Universal Press
2. Data Structures and Algorithm Analysis in C by Mark Alan Weiss, 2nd ed., Pearson Education
3. Data Structues and Algorithms by Aho, Hopcroft & Ullman
4. Data Structures and Program Design by Kruse et. al., PHI
5. Data Structures using C and C++ by Tanenbaum et. al., PHI
6. Algorithms + Data Structures = Programs by N. Wirth, PHI

COS-102: Advanced Computer Architecture

Course Outcomes :

On successful completion of this course, students will be able to

- define the principles of Parallel Algorithm Design.
- understand the fundamental concepts, techniques in Parallel Computation Structuring and Design.
- solve the algorithms using Parallel Programming Principle
- distinguish various architectures of high-performance computing systems.
- interpret modern design structures of pipelined and multiprocessors systems.

Introduction: Computer Architecture & Organization. Basic Parallel Processing Architecture, Taxonomy SISD, MISD, SIMD, MIMD structures, Serial, Parallel & Concurrent Computation, CISC vs RISC, Structure of Instruction of instruction sets and Desirable Attributes.
Pipelining: Basic Concepts of pipelining, Instruction Pipelining. Hazards, Reservation Tables, Collision, Latency, Dynamic pipeline, Vector processing & Vector processors.
Memory Systems: Cache Memory & Virtual Memory: Structure, Analysis & Design.

I/O Systems: Design Issues, Performances Measures.
Multiprocessor Architecture: Loosely Coupled & Tightly Coupled Systems, Concurrency & Synchronization, Scalability, Models of Consistency, and Application of SIMD Structure.
Interconnection Network: Definition. Types of Interconnected Networks; Baselines, Shuffle-Exchange, Omega, Cuba, Comparison & Application.

Systolic Architecture: Mapping Algorithm to array structures, Systolic processors. Mapping design & Optimization, Wave Front Array processor.
Data Flow Architecture: Data Flow Graphs, Petri nets, Static & Dynamic DFA.
Programming Environment: Different Models, Languages, Compilers, dependency Analysis. Message Passing, Program mapping to Multiprocessors, Synchronization.
Case Study: Basic Features of Current Architectural Trends. DSP Processor, Dual core Technology

References:
1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
3. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.

COS-103: Data Communication and Computer Network

Course Outcomes :

On successful completion of this course, students will be able to

- Understand computer network basics, network architecture, TCP/IP and OSI reference models.
- Identify and understand various techniques and modes of transmission.
- Describe data link protocols, multi-channel access protocols and IEEE 802 standards for LAN
- Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme.
- Discuss the elements and protocols of transport layer.
- Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

Overview of data communication and Networking: Introduction; Data communications: components, Direction of data flow (simplex, half duplex, full duplex); Networks: distributed processing, network criteria, Topology, categories of network (LAN, MAN, WAN); Protocols and standards; Reference models: OSI reference model, TCP/IP reference model, their comparative study.
Physical Layer: Data and signal fundamentals, Transmission impairments, Data rate limits for noisy and noiseless channels, Different line coding schemes, Block Analog to digital encoding, Analog Transmission, Concept of multiplexing, Frequency division multiplexing, Time division multiplexing
Data link layer: Types of errors, error detection & correction methods; Flow control; Protocols: Stop & wait ARQ, Go-Back- N ARQ, Selective repeat ARQ, HDLC; Multiple Access- Random Access, Controlled Access, Channelization
Network Layer: Introduction, IP Addressing, Routing Algorithms- Non Adaptive and Adaptive, Routers, ICMP, IPv6
Transport Layer: Introduction to Transport Layer Services, Connectionless Transport: UDP, Connection Oriented Transport: TCP, Congestion Control, Sockets, Quality of services
Application layer: DNS; SMTP, SNMP, FTP, HTTP & WWW, user authentication, Firewalls.
References:
1. William Stallings, Data and Computer Communication, Prentice Hall of India.
2. Behrouz A. Forouzan, Data Communication and Networking, McGraw-Hill.
3. AndrewS. Tanenbaum, Computer Networks, Prentice Hall.

COS-104: Software Engineering and project management

Course Outcomes :

On successful completion of this course, students will be able to

- Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice.
- Able to use modern engineering tools necessary for software project management, time management and software reuse.
- Plan a software engineering process life cycle, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements.
- Able to elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of the project
- Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology.

Introduction to software engineering and project management: Introduction to Software Engineering: Software, Evolving role of software, Three "R"-Reuse, Reengineering and Retooling, An Overview of IT Project Management: Define project, project management framework, The role of project Manager, Systems View of Project Management, Stakeholder management, Project phases and the project life cycle.
Software Process Models : Waterfall Model, Evolutionary Process Model: Prototype and Spiral Model, Incremental Process model: Iterative approach, RAD, JAD model, Concurrent Development Model, Agile Development: Extreme programming, Scrum.
Software Requirement Analysis and Specification : Types of Requirement, Feasibility Study, Requirement Analysis and Design: DFD, Data Dictionary, HIPO Chart, Warnier Orr Diagram, Requirement Elicitation: Interviews, Questionnaire, Brainstorming, Facilitated Application Specification Technique (FAST), Use Case Approach. SRS Case study, Software Estimation: Size Estimation: Function Point (Numericals). Cost Estimation: COCOMO (Numericals), COCOMO-II (Numericals). Earned Value Management.
Software Project Planning : Business Case, Project selection and Approval, Project charter, Project Scope management: Scope definition and Project Scope management, Creating the Work Breakdown Structures, Scope Verification, Scope Control.
Project Scheduling and Procurement management : Relationship between people and Effort: Staffing Level Estimation, Effect of schedule Change on Cost, Degree of Rigor & Task set selector, Project Schedule, Schedule Control, CPM (Numericals), Basic Planning Purchases and Acquisitions, Planning Contracting, Requesting Seller Responses, Selecting Sellers, Out Sourcing: The Beginning of the outsourcing phenomenon, Types of outsourcing relationship, The realities of outsourcing, Managing the outsourcing relationship.
Software Quality : Software and System Quality Management: Overview of ISO 9001, SEI Capability Maturity Model, McCalls Quality Model, Six Sigma, Formal Technical Reviews, Tools and Techniques for Quality Control, Pareto Analysis, Statistical Sampling, Quality Control Charts and the seven Run Rule. Modern Quality Management, Juran and the importance of Top management, Commitment to Quality, Crosby and Striving for Zero defects, Ishikawa and the Fishbone Diagram.
Human Resource Management : Human Resource Planning, Acquiring the Project Team: Resource Assignment, Loading, Leveling, Developing the Project Team: Team Structures, Managing the Project Team, Change management: Dealing with Conflict & Resistance Leadership & Ethics.
Software Risk Management and Reliability issues : Risk Management: Identify IT Project Risk, Risk Analysis and Assessment, Risk Strategies, Risk Monitoring and Control, Risk Response and Evaluation. Software Reliability: Reliability Metrics, Reliability Growth Modeling.

References:

1. Software Engineering, 5th and 7th edititon, by Roger S Pressman, McGraw Hill publication.
2. Managing Information Technology Project, 6edition, by Kathy Schwalbe, Cengage Learning publication.
3. Information Technology Project Management by Jack T Marchewka Wiley India publication.
4. Software Engineering 3rd edition by KK Agrawal, Yogesh Singh, New Age International publication.
5. Software Engineering Project Management by Richard H. Thayer Wiley India Publication
6. Software Engineering for students: A Programming Approach by Douglas Bell, Pearson publication.

COS-191: Data Structure Lab

Course Outcomes :

On successful completion of this course, students will be able to

- implement the recall the fundamentals of data structure with their implementation and its applications.
- implement the infer the complexity of algorithms.
- implement the apply the Greedy methods to solve real time problems.
- implement the compare different sorting and searching techniques.
- implement the design new algorithms with Dynamic Programming Techniques for Analytical Problems.

Experiments should include but not limited to :
Implementation of array operations:
Stacks and Queues: adding, deleting elements Circular Queue: Adding & deleting elements Merging Problem : Evaluation of expressions operations on Multiple stacks & queues :
Implementation of linked lists: inserting, deleting, inverting a linked list.
Implementation of stacks & queues using linked lists: Polynomial addition, Polynomial multiplication
Sparse Matrices : Multiplication, addition.
Recursive and Nonrecursive traversal of Trees
Threaded binary tree traversal.
AVL tree implementation Application of Trees.
Application of sorting and searching algorithms
Hash tables implementation: searching, inserting and deleting, searching & sorting techniques.

References:

1. Fundamentals of Data Structures in C by Horowitz, Sahni & Anderson-Freed, 2e Universal Press
2. Data Structures and Algorithm Analysis in C by Mark Alan Weiss, 2nd ed., Pearson Education
3. Data Structures using C and C++ by Tanenbaum et. al., PHI

COS-192: Network Programming Lab

Course Outcomes :

On successful completion of this course, students will be able to

- Practice the Understand computer network basics, network architecture, TCP/IP and OSI reference models.
- Practice the Identify and understand various techniques and modes of transmission.
- Practice the Describe data link protocols, multi-channel access protocols and IEEE 802 standards for LAN
- Practice the Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme.
- Practice the Discuss the elements and protocols of transport layer.

- Practice the Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

Problems and assignments based on Paper MSc-103
(Introduction to Sockets, Creating and Destroying, Specifying Addresses, TCP Client, TCP Server, UDP Client, UDP Server, Socket Options, Signals, Multitasking, Multiplexing, Multiple Recipients. Mapping Between Names and Internet Addresses, Finding Service Information by Name)

COS-201: Advanced Database Management System
Course Outcomes :

On successful completion of this course, students will be able to

- Describe DBMS architecture, physical and logical database designs, database modeling, relational, hierarchical and network models.
- Identify basic database storage structures and access techniques such as file organizations, indexing methods including B‐tree, and hashing.
- Learn and apply Structured query language (SQL) for database definition and database manipulation.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Understand various transaction processing, concurrency control mechanisms and database protection mechanisms.

Introduction : Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.
Relational Databases: Integrity Constraints revisited: Functional, Multi-valued and Join Dependency, Template Algebraic, Inclusion and Generalized Functional Dependency, Chase Algorithms and Synthesis of Relational Schemes. Query Processing and Optimization: Evaluation of Relational Operations, Transformation of Relational Expressions, Indexing and Query Optimization, Limitations of Relational Data Model, Null Values and Partial Information.
Object-oriented Databases - Objects and Types, Specifying the behavior of objects, Implementing Relationships, Inheritance. Sample Systems. New Database Applications.
Multimedia Database - Multimedia and Object Oriented Databases, Basic features of Multimedia data management, Data Compression Techniques, Integrating conventional DBMSs with IR and Hierarchical Storage Systems, Graph Oriented Data Model, Management of Hypertext Data, Client Server Architectures for Multimedia Databases.
Deductive Databases:Datalog and Recursion, Evaluation of Datalog program, Recursive queries with negation. Objected Oriented and Object Relational Databases: Modeling Complex Data Semantics, Specialization, Generalization, Aggregation and Association, Objects, Object Identity, Equality and Object Reference, Architecture of Object Oriented and Object Relational Databases. Case Studies: Gemstone, O2, Object Store, SQL3, Oracle xxi, DB2.
Parallel and Distributed Databases: Distributed Data Storage: Fragmentation and Replication, Location and Fragment Transparency, Distributed Query Processing and Optimization, Distributed Transaction Modeling and Concurrency Control, Distributed Deadlock, Commit Protocols, Design of Parallel Databases, Parallel Query Evaluation.

Advanced Transaction Processing: Nested and Multilevel Transactions, Compensating Transactions and Saga, Long Duration Transactions, Weak Levels of Consistency, Transaction Work Flows, Transaction Processing Monitors. Active Databases: Triggers in SQL, Event Constraint and Action:

ECA Rules, Query Processing and Concurrency Control, Compensation and Databases Recovery. WEB Databases: Accessing Databases through WEB, WEB Servers, XML Databases, commercial Systems: Oracle xxi, DB2. Data Mining: Knowledge Representation Using Rules, Association and Classification Rules, Sequential Patterns, Algorithms for Rule Accessing.

References:

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. J. D. Ullman, Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe, Fundamentals of Database Systems8, Addison-Wesley.
6. Serge Abiteboul, Richard Hull and Victor Vianu, Foundations of Databases. Addison-Wesley.

COS-202:M1: Automata Theory M2: Compiler Construction

Course Outcomes :

On successful completion of this course, students will be able to

- Understand the basic concepts of formal languages, automata and grammar types, as well as the use of formal languages and reduction in normal forms
- Demonstrate the relation between regular expressions, automata, languages and grammar with formal mathematical methods
- Design push down automata, cellular automata and turing machines performing tasks of moderate complexity
- Analyze the syntax and formal properties, parsing of various grammars such as LL(k) and LR(k)
- Describe the rewriting systems and derivation languages
- Define the Loop Optimization and DAG for source code.
- Explain the data structures for Block Structured Languages.
- Apply various parsing and conversion techniques for the design of a compiler.
- Analyze the concept of parsing techniques.
- Evaluate the Code Optimization and code generation techniques.

Automata Theory

Finite State Machines : Definition, concept of sequential circuits, state table & state assignments, concept of synchronous, asynchronous and liner sequential machines.
Finite State Models : Basic definition, mathematical representation, Moore versus Mealy m/c, capability & limitations of FSM, state equivalence & minimization, machine equivalence,

incompletely specified machines, merger graph & compatibility graph, merger table, Finite memory, definite, information loss less & inverse machines : testing table & testing graph.
Structure of Sequential Machines : Concept of partitions, closed partitions, lattice of closed partitions, decomposition : serial & parallel.
Finite Automation : Preliminaries (strings, alphabets & languages, graphs & trees, set & relations), definition, recognition of a language by an automata - idea of grammar, DFA, NFA, equivalence of DFA and NFA, NFA with e-moves, regular sets & regular expressions : equivalence with finite automata, NFA from regular expressions, regular expressions from DFA, two way finite automata equivalence with one way, equivalence of Moore & Mealy machines, applications of finite automata.
Closure Properties of Regular Sets : Pumping lemma & its application, closure properties minimization of finite automata : minimization by distinguishable pair, Myhill-Nerode theorem.

Context Free Grammars : Introduction, definition, derivation trees, simplification, CNF & GNF.
Pushdown Automata : Definition, moves, Instantaneous Descriptions, language recognised by PDA, deterministic PDA, acceptance by final state & empty stack, equivalence of PDA and CFL.
Closure Properties of CFLs : Pumping lemma & its applications, ogden's lemma, closure properties,
decision algorithms.
Introduction to Z. Regular language properties and their grammars. Context sensitive languages.

References:

1. K.L.P Mishra & N. Chandrasekharan- "Theory of Computer Science", PHI
2. Hopcroft JE. and Ullman JD., "Introduction to Automata Theory, Languages & Computation", Narosa.
3. Ash & Ash- "Discrete Mathematics",TMH
4. Martin-Introduction
5. Lewis H. R. and Papadimitrou C. H., "Elements of the theory of Computation", P.H.I.
6. Kain, "Theory of Automata & Formal Language", McGraw Hill.
7: Kohavi ZVI, "Switching & Finite Automata", 2nd Edn., Tata McGraw Hill.
8. Linz Peter, "An Introduction to Formal Languages and Automata", Narosa
9. "Introduction to Formal Languages", Tata McGraw Hill, 1983.

Compiler Construction

Introduction to Compiler, Different phases and passes of compiler.
Lexical Analysis: Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Finite state machines and regular expressions and their applications to lexical analysis, Implementation of lexical analyzers

Syntax Analysis: Role of the parser, Formal grammars and their application to syntax analysis, Context free grammars, Derivation and parse trees, Top Down parsing, LL(1) grammars, Predictive Parsing, Bottom-up-parsing, Shift Reduce Parsing, LR(0) grammars, LR parsing algorithms.
Syntax Directed Translation: Syntax directed definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions, L-attributed definitions.
Runtime Environments: Source Language issues, Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter passing mechanism.
Intermediate Code Generation: Intermediate languages, Graphical representation, Threeaddress code, Implementation of three address statements (Quadruples, Triples, Indirect triples).
Code Optimization and generation: Introduction, Basic blocks and flow graphs, Transformation of basic blocks, DAG representation of basic blocks, Principle sources of optimization, Loops in flow graph, Peephole optimization. Issues in the design of code generator, Register allocation and assignment.
Loader and Linkers: Basic Concepts of Linkers and Loader Functions, Boot Loaders, Linking Loaders, Linkage Editors, Dynamic Linking .
Concept of Editor and text editor, Interpreters, Simulator, Text editors - Overview of the Editing Process - User Interface – Editor Structure. – Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User Interface Criteria.

References:

1. Alfred Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman, "Compilers Principles, Techniques and Tools", Pearson Education Asia (2nd Ed. - 2009).
7. Leland L. Beck**,** "System Software: An Introduction to Systems Programming", 3/E, Addison-Wesley, 1997.
3. Allen I. Holub "Compiler Design in C", Prentice Hall of India, 2003.
4. C. N. Fischer and R. J. LeBlanc, "Crafting a compiler with C", Pearson Education.
5. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2003.
6. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI,

2001.
7. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thomson Learning.
8. Systems Programming and Operating Systems – D. M. Dhamdhere, TMH
9. John J. Donovan, “ Systems Programming”, 3rd edition, 1997, Addison Wesley.

COS-203:M1: OOPS using JAVA M2: Programming in R

Course Outcomes :

On successful completion of this course, students will be able to

- Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
- Understand dynamic memory management techniques using pointers, constructors, destructors, etc
- Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
- Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
- Demonstrate the use of various OOPs concepts with the help of program
- Demonstrate the use of various implementation of R program language.

OOPS using JAVA

Object Oriented System Development: Understanding Object Oriented Development, Understanding Object Oriented Concepts, Benefits of Object Oriented Development.
Java Programming Fundamentals: Introduction, Overview of Java, Data types, Variables and Arrays, Operators, Control Statements, Classes, Methods, Inheritance, Packages and Interfaces. Exceptional Handling, Multithreaded Programming, Reading console input and output, Reading and Writing Files, Print Writer Class, String Handling.

References:

1. Rambaugh, James Michael, Blaha – "Object Oriented Modelling and Design" – Prentice Hall, India

2. Patrick Naughton, Herbert Schildt – "The complete reference-Java2" – TMH

3. Deitel and Deitel – "Java How to Program" – 6th Ed. – Pearson

4. Ivor Horton's Beginning Java 2 SDK – Wrox

5. E. Balagurusamy – " Programming With Java: A Primer" – 3rd Ed. – TMH

Programming in R
History and overview of R.
Install and configuration of R programming environment.
Basic language elements and data structures.
R+Knitr+Markdown+GitHub.
Data input/output 6. Data storage formats.
Subsetting objects.
Vectorization.
Control structures

Functions
Scoping Rules
Loop functions
Graphics and visualization
Grammar of data manipulation (dplyr and related tools)
Debugging/profiling
Statistical simulation

COS-204: M1: Computer Fundamentals M2: Programming Concepts

Course Outcomes :

On successful completion of this course, students will be able to

- Bridge the fundamental concepts of computers with the present level of knowledge of the students.
- Familiarise operating systems, programming languages, peripheral devices, networking, multimedia and internet
- Understand binary, hexadecimal and octal number systems and their arithmetic.
- Understand how logic circuits and Boolean algebra forms as the basics of digital computer.
- Understanding a functional hierarchical code organization.
- Ability to define and manage data structures based on problem subject domain.
- Ability to work with textual information, characters and strings.
- Ability to work with arrays of complex objects.
- Understanding a concept of object thinking within the framework of functional model.
- Understanding a concept of functional hierarchical code organization.
- Understanding a defensive programming concept. Ability to handle possible errors during program execution.

Computer Fundamentals
Introduction to Computers, Data representation, Conversion of data. Memory organization, Different secondary storage devices and Magnetic media devices.
Data Representation: Representation of Characters in Computers, Representation of Integers, Representation of Fractions, Hexadecimal Representation of Numbers, Decimal to Binary Conversion, Error Detecting Codes
Basic concepts of Programming, Machine code, Assembly Language (Introduction), Problem analysis, program constructions – flowcharts, algorithms, pseudo codes, data structures – stacks, queues, linked lists etc., approaches to programming – top-down, bottom-up approach, divide & conquer, modular programming.

Programming Concepts
Preliminaries, Constants & Variables, Arithmetic Expressions, Input Output statements, Control Statements, Do-Statements, C-Preprocessor, Do-While statement, if-else statement, Array, Pointer.
Elementary Format Specifications, Logical Statements & Decision Tables, Function & Subroutines, handling of arrays, matrices, handling of character strings
References:
1. Yashavant P. Kanetkar, Let Us C, BPB Publications.
2. Balagurusamy, Programming in ANSI C, Mcgraw Hill Education.
3. B. W. Kernighan & D. M. Ritchie, C Programming Language

COS-291: DBMS Lab

Course Outcomes :

On successful completion of this course, students will be able to

- connect a java program to a DBMS and perform insert, update and delete operations on DBMS table.
- develop a dynamic webpage by the use of java script and HTML/ DHTML/ASP.
- write a well formed / valid XML document
- write a server side java application called Servlet to catch form data sent from client and store it on database.
- write a server side java application called JSP to catch form data sent from client, process it and store it on database.

Structured Query Language

1. Creating Database
Creating a Database
Creating a Table
Specifying Relational Data Types
Specifying Constraints
Creating Indexes
2. Table and Record Handling
INSERT statement
Using SELECT and INSERT together
DELETE, UPDATE, TRUNCATE statements
DROP, ALTER statements
3. Retrieving Data from a Database
The SELECT statement
Using the WHERE clause
Using Logical Operators in the WHERE clause
Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING Clause
Using Aggregate Functions
Combining Tables Using JOINS
Subqueries
4. Database Management
Creating Views
Creating Column Aliases
Creating Database Users
Using GRANT and REVOKE
Cursors in Oracle PL / SQL
Writing Oracle PL / SQL Stored Procedures
5. Use of user interfaces and report generation utilities typically available with RDBMS products.

References

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. SQL, PL/SQL the Programming Language of Oracle by Ivan Bayross, Paperback

COS-292: M1: OOPS Lab M2: R Lab
Course Outcomes :

On successful completion of this course, students will be able to

- design and implement programs in the Java programming language that make strong use of classes and objects.
- print formatted text to the console output and read/parse console input text using a Scanner object.
- apply logical constructs for branching and loops as well as use integrator objects when appropriate.
- define classes and methods.
- implement R programming logic.

OOPS Lab
1. Assignments on class, constructor, overloading, inheritance, overriding
2. Assignments on wrapper class, vectors, arrays
14
3. Assignments on use of use of abstract class
4. Assignments on developing interfaces- multiple inheritance, extending interfaces
5. Assignments on creating and accessing packages
6. Assignments on multithreaded programming, handling errors and exceptions, applet programming and graphics programming

R Lab
Vectors and matrix operations
Introduction to data and visualization
Functions
Different Simulated evolution
Strings and regular expressions
Shaping data and using plyr
Exploring large data sets: eg. US baby names
Tidying and reshaping data
A small data project

COS-301: Advanced operating System
Course Outcomes :

On successful completion of this course, students will be able to

- Describe the architecture and features of UNIX Operating System and distinguish it from other Operating System
- Demonstrate UNIX commands for file handling and process control
- Write Regular expressions for pattern matching and apply them to various filters for a specific task
- Analyze a given problem and apply requisite facets of SHELL programming in order to devise a SHELL script to solve the problem

Introduction: Introduction to Operating Systems, Concept of batch-processing, multi-programming, time sharing, real time operations
Process Management: Concept of process, state diagram, process control block; scheduling of processes criteria, types of scheduling, non-preemptive and preemptive scheduling algorithms
Threads: Concept, process vs thread, kernel and user threads, multithreading models
Inter-process Communication (IPC): Shared memory, message, FIFO, concept of semaphore, critical region, monitor
Process Synchronization: concepts, race condition, critical section problem and its solutions; synchronization tools-semaphore, monitor etc., discussion of synchronization problems. Deadlock – conditions, resource allocation graph, prevention techniques, avoidance technique –Banker's algorithm and related algorithms
Memory management: Address space and address translation; static partitioning, dynamic partitioning, different types of fragmentation, paging, segmentation, swapping, virtual memory, demand paging, page size, page table, page replacement algorithms, thrashing, working set strategy
File Management: File and operations on it, file organization and access; file allocation; directory structures, file sharing, file protection
Device management: Magnetic disks, disk scheduling-criteria and algorithms , disk management – formatting, boot block, disk free space management techniques, concept of RAID etc
Protection and Security: Concepts of domain, Access matrix and its implementation, access control, Security of systems-concepts, threats-Trojan horse, virus, worms etc, introduction to cryptography as security tool, user authentication
Case Studies

References:
1.Operating Systems Concepts –A. Silberschatz, P. Galvin and G. Gagne. Wiley India
2.Operating SystemsConcepts -Gary Nutt, N. Chaki and S. Neogy, Pearson Education
3.Operating Systems –W. Stallings, Pearson Education
4.Operating Systems: A Concept-based Approach –D. M. Dhamdhere, Tata McGraw-Hil

COS-302:M1: Computer Graphics M2: Image Processing
Course Outcomes :

On successful completion of this course, students will be able to

- Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
- Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
- Use of geometric transformations on graphics objects and their application in composite form.
- Extract scene with different clipping methods and its transformation to graphics display device.
- Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
- Render projected objects to naturalize the scene in 2D view and use of illumination models for this.
- Review the fundamental concepts of a digital image processing system.
- Analyze images in the frequency domain using various transforms.
- Evaluate the techniques for image enhancement and image restoration.
- Categorize various compression techniques.
- Interpret Image compression standards.
- Interpret image segmentation and representation techniques.

Computer Graphics

Introduction to computer graphics & graphics systems: Overview of computer graphics, representing pictures, preparing, presenting & interacting with pictures for presentations; Visualization & image processing; RGB color model, direct coding, lookup table; storage tube graphics display, Raster scan display, 3D viewing devices, Plotters, printers, digitizers, Light pens etc.; Active & Passive graphics devices; Computer graphics software.

Scan conversion: Points & lines, Line drawing algorithms; DDA algorithm, Bresenham's line algorithm, Circle generation algorithm; Ellipse generating algorithm; scan line polygon, fill algorithm, boundary fill algorithm, flood fill algorithm.

2D transformation & viewing: Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, transformations between coordinate systems; reflection shear; Transformation of points, lines, parallel lines, intersecting lines. Viewing pipeline, Window to viewport co-ordinate transformation, clipping operations, point clipping, line clipping, clipping circles, polygons & ellipse.

3D transformation & viewing: 3D transformations: translation, rotation, scaling & other transformations. Rotation about an arbitrary axis in space, reflection through an arbitrary plane; general parallel projection transformation; clipping, viewport clipping, 3D viewing.

Curves : Curve representation, surfaces,designs, Bezier curves, B-spline curves, end conditions for periodic B-spline curves, rational B-spline curves.

Hidden surfaces: Depth comparison, Z-buffer algorithm, Back face detection, BSP tree method, the Printer's algorithm, scan-line algorithm; Hidden line elimination, wire frame methods, fractal - geometry.

Color & shading models: Light & color model; interpolative shading model; Texture;

References:

1. Hearn, Baker – – Computer Graphics (C version 2nd Ed.)‖ – Pearson education
2. Z. Xiang, R. Plastock – – Schaum's outlines Computer Graphics (2nd Ed.)‖ – TMH
3. D. F. Rogers, J. A. Adams – – Mathematical Elements for Computer Graphics (2nd Ed.)‖ – TMH
4. Mukherjee, Fundamentals of Computer graphics & Multimedia, PHI
5. Mukherjee Arup, Introduction to Computer Graphics

Image Processing Lectures

Fundamentals of Digital Image Processing, Image representation, Basic Image transforms, image file format .

Image Enhancement: Contrast stretching, Histogram Equalization, Binarization

Filtering in Spatial domain: Mean filter, Order Statistics filters.

Filtering in Frequency domain : Butterworth filter, Gaussian filter.

Image Restoration : Image degradation models, Weiner filter.

Image textures: Run Length Coding, Gray-level co-occurrence matrix

Image Segmentation: Edge detection: Gradient operators, Compass operator, Laplacian operators. LoG operator.

Region Segmentation : Region growing, region splitting and merging.

Shape detection: Least Mean Square error line fitting, Eigenvector line fitting, Straight line Hough Transform, Generalized Hough Transform.

Morphological Operators: Dilation, Erosion, Opening , Closing, Hit-and-Miss transforms, Applications.

Image Compression.

Image Understanding: Feature extraction techniques, Statistical Decision making techniques, Nearest Neighbour Clustering, Maxi-min Clustering, Discriminant functions, Artificial Neural Networks.

References:

1. Digital Image Processing, Gonzalves,Pearson
2. Digital Image Processing, Jahne, Springer India
3. Digital Image Processing & Analysis,Chanda & Majumder,PHI
4. Fundamentals of Digital Image Processing, Jain, PHI
5. Image Processing, Analysis & Machine Vision, Sonka, VIKAS

COS-303: Elective – I

Course Outcomes :

On successful completion of this course, students will be able to

- Describe important types of combinatorial optimization problems
- Formulate combinatorial optimization problems as mathematical models and determine the difficulty of the problems with the help of complexity theory
- Explain the design of and the principles behind efficient solution methods and use the methods for solving combinatorial optimization problems
- Use available software for solving optimization problems take part of development of software for optimization problems

Graph Theory
Introduction

Graphs – Introduction – Isomorphism – Sub graphs – Walks, Paths, Circuits –Connectedness – Components – Euler graphs – Hamiltonian paths and circuits – Trees – Properties of trees – Distance and centers in tree – Rooted and binary trees.

Trees, Connectivity & Planarity

Spanning trees – Fundamental circuits – Spanning trees in a weighted graph – cut sets – Properties of cut set – All cut sets – Fundamental circuits and cut sets – Connectivity and separability – Network flows – 1-Isomorphism – 2-Isomorphism – Combinational and geometric graphs – Planer graphs – Different representation of a planer graph.

Matrices, Colouring And Directed Graph

Chromatic number – Chromatic partitioning – Chromatic polynomial – Matching – Covering – Four color problem – Directed graphs – Types of directed graphs – Digraphs and binary relations – Directed paths and connectedness – Euler graphs.

References:

1. NarasinghDeo, Graph theory, PHI.
2. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd.
3. Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.

Distributed Computing

Introduction to distributed environment: Goals, hardware & software concepts, P2P, Cluster, Grid, Cloud, the client-server model, Strengths and weakness of distributed computing, forms of computing
Communication: Layered protocols, RPC, remote object invocation, message-oriented communication
Distributed computing paradigms: Message passing, client server, P2P, remote procedure call model, distributed objects, object space, collaborative application (groupware)
Socket: Socket metaphor, datagram socket API, stream mode socket API, sockets with non blocking I/O, secure socket API
Java RMI: Client side, Server Side, object registry, Remote Interface, Server side software, client side

software, RMI vs Socket
Advanced RMI: Client callback, stub downloading, RMI security manager
Group Communication: Unicasting, multicasting, connection oriented & connectionless, reliable and unreliable multicast, Java basic multicast API
Internet Applications: HTML, XML, HTTP, Applets, Servlets, Web services, SOAP
Mobile Agents: Basic architecture, advantages, mobile agent framework systems, design, implementation using Java RMI
Distributed coordination-based systems JINI: Runtime environment, architecture, discovery protocol, join protocol, lookup service, distributed event, distributed leasing, transactions, surrogate architecture
New paradigms of distributed computing environment

References:

1. Distributed Computing: Principles and Applications, M. L. Liu, Pearson/Addison-Wesley.
2. A Programmer's Guide to Jini Technology, Jan Newmarch, Apress.
3. A. Taunenbaum, Distributed Systems: Principles and Paradigms, PHI
4. G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, Pearson Education
5. Core Jini, W. Kieth Edwards, Apress.

Mobile Computing

Introduction: Introduction to wireless networks and mobile computing – Characteristics, Issues and challenges.
Wireless Transmission: Fundamentals of wireless transmission - Medium Access Control Protocols, Different types of multiple access techniques and their characteristics.
Cellular Communication: Cellular concept, Overview of different Generations.
Mobile: Mobile IP, Mobile transport layer - Mechanisms for improving TCP performances on wireless links, , Overview of Security in mobile environments.

Wireless: Overview of Wireless LAN IEEE 802.11 series, Overview of Bluetooth, Overview of Wireless Sensor Networks.
Wireless application Environments: WAP, WML, Push Architecture, Push/Pull Services Mobile Adhoc Networks – Characteristics, Routing protocols.

References:

1. Mobile Computing, Raj Kamal, Oxford
2. Hansmann, Merk, Nicklous, Stober, "Principles of Mobile Computing", Springer, second edition, 2003.
3. Mobile Communications, Jochen Schiller, Pearson Education
4. Stojmenovic and Cacute, "Handbook of Wireless Networks and Mobile Computing", Wiley, 2002, ISBN 0471419028.

Pattern Recognition

Basic concepts of Pattern Recognition.
Pattern Preprocessing and Feature Selection.
Decision Functions.
Bayesian decision theory.
Parametric Estimation: Maximum likelihood estimation and Bayesian estimation.
Non- parametric Estimation: Parzen windows, Nearest Neighbor estimation.
Pattern Classification: Linear classifier: Perceptron, SVM.
Non-linear classifiers: MLP, Non-linear SVM.

Unsupervised learning and Clustering: Partitioning method, Density-based method, MSTbasedmethod, Self organizing map, Hierarchical Clustering, Cluster validity.
Syntactic Pattern Recognition (Basic concepts).
Some real-life applications Pattern Recognition

References:

1. Pattern Recognition Principles, Tou and Gonzalez, Addison-Wesley
2. Pattern Classification, Duda, Hart and Stork, Second Edition, Wiley
3. Pattern Recognition and Machine Learning, Christopher Bishop, Springer
4. Introduction to Statistical Pattern Recognition, Fukunaga, Second Edition, Academic Press

Machine Learning

Introduction:Machine learning applications, concepts learning Introduction to Bayesian learning theory: regression, feature selection, supervised learning, class
conditional probability distributions, Examples of classifiers Bayes optimal classifier and error, learning classification approaches, handling continuous attributes.
Decision tree learning algorithms: Inference model, general domains, symbolic decision trees, consistency, learning trees from training examples, entropy, mutual information, ID3 algorithm criterion, C4.5 algorithm, handling continuous and missing attributes, confidence, overfitting, pruning, learning with incomplete data
Artificial Neural Network: Single layer neural network, linear reparability, general gradient descent, perceptron learning algorithm, multi-Layer perceptron: two-layers universal approximators, backpropagation learning, important parameters, Margin of a classifier, dual perceptron algorithm, learning nonlinear hypotheses with perceptron.
Instance-based Learning: Nearest neighbor classification, k-nearest neighbor, nearest neighbor error probability,
Machine learning concepts and limitations: Learning theory, formal model of the learnable, sample complexity, learning in zero-bayes and realizable case, VC-dimension, fundamental algorithm independent concepts, hypothesis class, target class, inductive bias, occam's razor, empirical risk, limitations of inference machines, approximation and estimation errors, Tradeoff.
Support Vector Machine (SVM): Kernel functions, implicit non-linear feature space, theory, zero-Bayes, realizable infinite hypothesis class, finite covering, margin-based bounds on risk, maximal margin classifier.
Machine learning assessment and Improvement: Statistical model selection, structural risk minimization, bootstrapping, bagging, boosting.
Unsupervised learning: introduction, K- means clustering,
Hierarchical clustering Semi-supervised learning: introduction, self-training, co-training.

References:

1. T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.
2. E. Alpaydin, Introduction to Machine Learning, Prentice Hall of India, 2006.
3. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
4. R. O. Duda, P. E. Hart, and D.G. Stork, Pattern Classification, John Wiley and Sons, 2001.
5. Vladimir N. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
6. Shawe-Taylor J. and Cristianini N., Cambridge, Introduction to Support Vector Machines, University Press, 2000.

Soft Computing

Introduction to Soft Computing, Components of Soft Computing, Importance of Soft Computing, Applications.

Fuzzy Set Theory - Definition, Different types of fuzzy set membership functions. Fuzzy set theoretic operations, Fuzzy rules and fuzzy reasoning, Fuzzy inference systems.

Rough set theory.

Probabilistic Reasoning.

Genetic Algorithms, Simulated Annealing, applications.

Neural Networks- Artificial neural networks models, Supervised Learning, Unsupervised Learning, Applications.

Hybrid Systems and applications.

References:

1. Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence - Jang, Sun and Mizutani, Printice Hall.

2. Soft Computing : Integrating Evolutionary, Neural, and Fuzzy Systems, by Tettamanzi, Andrea, Tomassini, and Marco. (2001), Springer.

Recent Computing Architecture Trends

Distributed computing architectures: Introduction, Client-server architecture, Peer-to-peer systems, Applications.

Parallel and scalable architectures: Multiprocessors and multicomputer architectures, Multivector and SIMD computers, Scalable, multithreaded and data flow architectures.

Grid computing architectures: Introduction, Benefits, terms and concepts, grid user roles, grid architecture considerations, standards for grid environments, applications.

Cloud computing architectures: Introduction, Layers of cloud architecture, understanding cloud ecosystem, cloud architecture components, applications.

References:

1. Cloud Computing for Dummies by Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper (Wiley India Edition)

2. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.

3. *Distributed Computing.* Principles, Algorithms, and. Systems. Ajay D. Kshemkalyani. Mukesh Singhal. Cambridge University press

4. Principles of Grid Computing, Krishna, Paperback

COS-304: M1: DBMS M2: Internet Technology (CBCS)

Course Outcomes :

On successful completion of this course, students will be able to

- Describe DBMS architecture, physical and logical database designs, database modeling, relational, hierarchical and network models.
- Identify basic database storage structures and access techniques such as file organizations, indexing methods including B - tree, and hashing.
- Learn and apply Structured query language (SQL) for database definition and database manipulation.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Understand various transaction processing, concurrency control mechanisms and database protection mechanisms.
- develop a dynamic webpage by the use of java script and DHTML.
- write a well formed / valid XML document.
- connect a java program to a DBMS and perform insert, update and delete operations on DBMS table.

DBMS

Concept & Overview of DBMS.
Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.
Integrity Constraints: Functional, Multi-valued and Join Dependency, Inclusion and Generalized Functional Dependency.
ER Diagram Some real-life applications

Internet Technology

Computer Networks: What is Network, Network Structure, Reference models: OSI reference model, TCP/IP reference model
Interconnecting LAN segments: Internetworking devices- Hubs, Bridges, Switches, Routers, Gateways
Network access & physical media: Traditional Ethernet, Concept of Wireless LAN, Bluetooth & Wi-Fi
IP Addressing: classful addressing, Subnetting and super-netting, Masking, classless addressing
Internet services: Internet basics, services offered by internet, internet vs. intranet, WWW, SMTP, FTP, Telnet etc.
HTML Programming Basics: HTML Elements, Attributes, Headings, Paragraphs, Formatting, Fonts, Styles, Links, Images, Tables, Lists, Forms, Frames, Iframes, Colors etc.

References:

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. Computer Networking: A Top-Down Approach Featuring the Internet, by James F. Kurose and Keith W. Ross, 5th Edition, Pearson Education, 2010
5. Data communication and Networking, by Behrouz A. Forouzan, 4th Edition, Tata McGraw-Hill, 2007
6. Computer Networks, by Andrew S. Tanenbaum, 4th Edition, Prentice Hall India, 2003

COS-391: M1: Graphics Lab M2: OS Lab

Course Outcomes :

On successful completion of this course, students will be able to

- Understand the fundamental concepts of mobile devices and types of mobile operating systems to know about comprehensive knowledge in the field of computer science.
- Extract scene with different clipping methods and its transformation to graphics display device.
- Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
- Implement various field of computer graphics and Operating system application

Graphics Lab

Experiments should include but not limited to :

List of Experiments:

1. Write a program to implement DDA algorithm.
2. Write a program to draw a specified figure supplied by Instructor.
3. Write a program to implement Bresenham's line algorithm.
4. What are the advantages of Bresenham's line algorithm over DDA algorithm.
5. Write a program to implement Midpoint circle gen

6. Write a program to implement Bresenham's circle generating algorithm.
7. Write a program to draw the specified figure supplied by Instructor
8. Write a program to draw the specified figure supplied by Instructor
9. Write a program to implement outline character.
10. Write a program to implement bitmap character.
11. Write a program to implement ellipse generating algorithm
12. Write a procedure to scan the interior of a specified ellipse into a solid color.
13. Write the Scan line filling algorithm.
14. Write a program to implement Line Clipping Algorithm using Cohen Sutherland Algorithm.
15. Write a program to implement Line Clipping Algorithm using Liang Barsky Algorithm.
16. Explain the Sutherland and Cohen subdivision algorithm for the line clipping.
17. Write a program to Implement Polygon Clipping Algorithm using Sutherland -Hodgman Algorithm.
18. Write a program to implement scaling on polygon.
19. Write a program to implement transferring on polygon.
20. Write a program to implement rotation on polygon.
21. Write a program to implement reflection on polygon.
22. Write a Program to implement set of Basic Transformations on Polygon i.e. Translation,Rotation and Scaling.
23. Write a program to implement set of Composite Transformations on Polygon i.e Reflection,Shear (X &Y), rotation about an arbitrary point.

OS Lab

Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).
Process: starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.
Semaphore: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).
POSIX Threads: programming with pthread functions(viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)
Inter-process communication: pipes(use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO)

References

1.UNIX: Concepts & Applications, Sumitava Das, TMH
2.Your UNIX –The Ultimate Guide, Sumitava Das, TMH
3.Design of UNIX Operating System,Maurice Bach, PHI
4.Learning the UNIX operating Systems,Peek,SPD/O'REILLY
5.Mastering UNIX/LINUX/Solaris Shell Scripting, Randal k. Michael, Wiley Dreamtech

COS-392: M1: Industrial Tour M2: Term Paper

Course Outcomes :

On successful completion of this course, students will be able to

- comprehend the state-of-the-art requirements of the Industry.
- apply critical thinking, reasoning and creative thinking for Software Design in an industry as an individual or as a part of a team.
- analyze the problem and provide Solution by Decision Making.
- develop Interpersonal, Communication and Presentation skills.
- build the modules for a specific problem.

Industrial Tour

An Industrial Visit would be organized by the department for not less than 3 days and not more than one week and students should submit a report on that tour which will be examined by a board of examiners to be nominated by the B.O.S.

Term Paper

Seminar topic will be assigned to individual student by the Head of the department at the beginning of the semester.

COS-401: Artificial Intelligence

Course Outcomes :

On successful completion of this course, students will be able to

- Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
- Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
- Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Demonstrate proficiency developing applications in an 'AI language', expert system shell, or data mining tool.
- Demonstrate proficiency in applying scientific method to models of machine learning.
- Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

Introduction: Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, Agents.
State Space Search: Depth First Search, Breadth First Search, DFID.
Heuristic Search: Best First Search, Hill Climbing, Beam Search, Tabu Search.
Randomized Search: Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.
Finding Optimal Paths: Branch and Bound, A*, IDA*, Divide and Conquer approaches, Beam Stack Search.
Problem Decomposition: Goal Trees, AO*, Rule Based Systems, Rete Net.
Game Playing: Minimax Algorithm, Alpha Beta Algorithm, SSS*.
Planning and Constraint Satisfaction: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graph plan, Constraint Propagation.
Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and backward chaining.

References:
1. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.
2. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.
3. John Haugeland, Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985.
4. Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.
5. Zbigniew Michalewicz and Da

COS-402: Elective – II

Course Outcomes :

On successful completion of this course, students will be able to

- develop a dynamic webpage by the use of java script and DHTML.
- write a well formed / valid XML document.
- connect a java program to a DBMS and perform insert, update and delete operations on DBMS table.
- write a server side java application called Servlet to catch form data sent from client, process it and store it on database.
- write a server side java application called JSP to catch form data sent from client and store it on database.

Web Technology

Introduction to the Web Technologies: Concept of WWW, Internet and WWW, HTTP Protocol: Request and Response, Web browser and Web servers. Web Security and Firewalls, Web Protocols: TCP, IP and HTTP, SMTP, POP3, FTP

HTML: Basics of HTML, Structure of HTML code, formatting and fonts, color, hyperlink, lists, tables, images, DOM (Programming Assignments based on above topics)

Style Sheets: Need for CSS, introduction to CSS, basic syntax and structure, Classes and Pseudo Classes, CSS tags for setting background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning etc. (Programming Assignments based on above topics)

Client Side scripting Language: (JavaScript/ VBScript etc.) and DHTML.

Introduction to PHP: Configuration and Installation of PHP, basic syntax of PHP, Expressions, Statements, Arrays, Functions, string, Regular Expressions, Date and Time Functions (Programming Assignments based on above topics)

PHP and MySQL: File Handling- Creating a File, Reading from Files, Copying Files, Moving File, Deleting File, Updating File, Uploading Files, Form Designing using HTML 5, Validation's using PHP Connection to server, creating database, selecting a database, listing database, listing table names, creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, Master-Detail relationships using Joins. Session Management- Using Cookies in PHP, HTTP Authentication, Using Sessions. (Programming Assignments based on above topics)

Web services: Design and modeling of web services, Technologies for implementing web services

References:

1. Web Technologies, Black Book, Dreamtech Press
2. Learning PHP, MySQL, JavaScript, CSS and HTML 5, Robin Nixon, O'Reilly publication
3. Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
4. Professional PHP Programming, Jesus Caspagnetto, Etal. Wrox Publication.

5. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson
6. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

Data Mining

Introduction: concepts of data mining,
Differences of conventional Database with data warehouse.
Concepts of Data Cubes and OLAP Data Processing.
Types of Data Warehouses and Schemas.
Development Methodologies.
Management of Data Warehouses.
Data Mining Algorithms in general with scalability issues.
Evaluation of data mining results.
Data Preprocessing Techniques.
Application of Association Rule Mining in data mining.
Application of Clustering Algorithms in data mining.
Application of Classification Algorithms in data mining.
Text mining, Web mining and other Applications.
Recent Trends.

References:

1. J. Han & M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2nd Ed.
2. Data warehousing: OLAP & data mining, S. Nagabhushan, New age publications.
3. Introduction to data mining by Tan, Steinbach, Kumar, Pearson Education
4. Data mining: A tutorial based primer by Roiger, Geatz,, Pearson Education

Parallel Computing

Introduction to High Performance Computing: Milestones and applications.
High-Performance Computing architectures: Overview of the major classes of HPC architectures and their evolution.
Parallel programming models and performance analysis: Parameterisation, modeling, performance analysis, Amdahl's law, efficiency, and benchmarking of systems.
Programming parallel computers: Overview of parallel programming, parallel languages, parallelizingcompilers, message passing and data parallel programming models, introduction to MPI and OpenMP.
Multi-Thread Models with primary sources of overhead, memory architecture and memory access times and associated sources of overhead; Multi-Process Execution Model. Restructuring for Parallel Performance - Loop Transformations; Data Transformations; Dependence Analysis; Compiler Strategies.
Parallel Algorithms - Cyclic Reduction; Iterative Algorithms (Jacobi, Gauss-Seidel and Red-BlackOrderings); Divide-and-Conquer Algorithms, Adaptive Quadrature.

References:

1. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey
2. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series,
3. Parallel Programming in C with MPI and OpenMP by M.J. Quinn, McGraw-Hill

Cryptography and Steganography

Introduction to security attacks, services and mechanism, introduction to cryptography. Conventional Encryption: Conventional encryption model, classical encryption techniques- substitution ciphers and transposition ciphers, cryptanalysis, stereography,
Stream and block ciphers.
Modern Block Ciphers: Block ciphers principals, Shannon's theory of confusion and diffusion, fiestal

structure, data encryption standard(DES), strength of DES, differential and linear crypt analysis of DES, block cipher modes of operations, triple DES, IDEA encryption and decryption, strength of IDEA, confidentiality using conventional encryption, traffic confidentiality, key distribution, random number generation.
Introduction to Information Hiding: Technical Steganography, Linguistic Steganography, Copy Right Enforcement, Wisdom from Cryptography Principles of Steganography: Framework for Secret Communication, Security of Steganography System, Information Hiding in Noisy Data , Adaptive versus non-Adaptive Algorithms, Active and Malicious Attackers, Information hiding in Written Text.
A Survey of Steganographic Techniques: Substitution systems and Bit Plane Tools, Transform Domain Techniques: - Spread Spectrum and Information hiding, Statistical Steganography, Distortion Techniques, Cover Generation Techniques. Steganalysis: Looking for Signatures: - Extracting hidden Information, Disabling Hidden Information.
Watermarking and Copyright Protection: Basic Watermarking, Watermarking Applications, Requirements and Algorithmic Design Issues, Evaluation and Benchmarking of Watermarking system.

References:

1. William Stallings, "Cryptography and Network Security: Principals and Practice", Prentice Hall, New Jersy.

2. Johannes A. Buchmann, "Introduction to Cryptography", Springer-Verlag.

3. Bruce Schiener, "Applied Cryptography".

4. Katzendbisser, Petitcolas, " Information Hiding Techniques for Steganography and Digital Watermarking", Artech House.

5. Peter Wayner, "Disappearing Cryptography: Information Hiding, Steganography and Watermarking 2/e", Elsevier

6. Bolle, Connell et. al., "Guide to Biometrics", Springer

Bioinformatics

Introduction to molecular biology, The Central Dogma of Molecular Biology, Physical mapping.
Protein sequence data bank. NBFR-PIR, SWISSPROT, GenBank, EMBL nucleotide sequence data bank, Protein Data Bank (PDB) etc.
Motif finding in DNA and proteins.
Sequence alignment for DNA and protein sequences, Concepts: homology, sequence similarity and sequence alignment; dynamic programming algorithms, Pairwise alignment, Global and local alignment using dynamic programming, Heuristic alignment methods: BLAST/FASTA and the statistics of local alignments, Multiple sequence alignment: Definition, scoring, techniques, Aligners for proteins sequences, Spliced alignment
Gene ontology, Annotation and Metadata.
Secondary and Tertiary Structure predictions; Chao-Fasman algorithms; The basic HMM algorithms: forward, backward, Viterbi, Baum-Welch; Neural Networking.

References:

[1] M. Lesk, "Introduction to Bio Informatics," Oxford University Press

[2] Hooman Rashidi, Lukas K. Buehler, "Bioinformatics Basics: Applications in Biological Science and Medicine," CRC Press/Taylor & Francis Group, 2nd edition, May 2005

[3] Jeffrey Augen, "Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and Information-Based Medicine," Addison-Wesley

[4] Stephen A. Krawetz, David D. Womble, "Introduction to Bioinformatics: A Theoretical and
Practical

Approach,'' Humana Press

[5] Bryan Bergeron, ''Bioinformatics Computing,'' Prentice Hall PTR

[6] Malcolm Campbell, Laurie J. Heyer, ''Discovering Genomics, Proteomics, and Bioinformatics,'' Benjamin/Cummings

Natural Language Processing

Speech & Natural Language Processing:
Brief Review of Regular Expressions and Automata; Finite State Transducers;
Word level Morphology and Computational Phonology;
Basic Text to Speech;
Introduction to HMMs and Speech Recognition.
Indian language case studies; Part of Speech Tagging;
Parsing with CFGs; Probabilistic Parsing.
Representation of Meaning;
Semantic Analysis; Lexical Semantics; Word Sense;
Disambiguation; Discourse understanding;
Natural Language Generation.

References:

1.Natural Language Processing And Information Retrieval, TANVEER SIDDIQUI, U. S TIWARY, Oxford University Press
2. NATURAL LANGUAGE UNDERSTANDING, J Allen, Pearson India
3. Multilingual Natural Language Processing Applications from Theory to Practice, Bikel, Pearson India
4. NATURAL LANGUAGE PROCESSING, Dipti Mishra Sharma, MACMILLAN INDIA LTD

COS-403: Elective – III

Course Outcomes :

On successful completion of this course, students will be able to

- Understand the Importance of IT and its acts in India.
- Understanding the basic concept of computer fundamentals and number systems
- Describe about the basic components of computer.
- Understand the applications of MS Word, MS Excel and MS Power Point in documentation and other areas.
- Understanding the concept of DBMS and its importance in record maintenance.
- Describe important types of combinatorial optimization problems
- Formulate combinatorial optimization problems as mathematical models and determine the difficulty of the problems with the help of complexity theory

Information Security

Introduction and Security Trends
General Security Concepts and introduction to what is an ''infosphere''
Inside the Security Mind
Operational Security and People's Role in Information Security
Cryptography
Internet Standards and Physical Security
Network Security and Infrastructure
Authentication and Wireless

Intrusion Detection Systems and Security Baselines
Attacks and E-mail
Web Security and Software Security
Disaster Planning and Risk Management
Change and Privilege Management
Computer Forensics and the Law
Privacy Issues and Review for Final

References:

1. P *rinciples of Computer Security: Security+ and Beyond*
Wm. Arthur Conking, Gregory B. White, *et al*
(McGraw Hill, 2010) ISBN: 978-0-07-163375-8

Embedded System

Introduction- Embedded system overview, embedded hardware units, embedded software in a system, embedded system on chip (SOC), design process, classification of embedded systems.
Embedded computing platform - CPU Bus, memory devices, component interfacing, networks for embedded systems, communication interfacings: RS232/UART, RS422/RS485, IEEE 488 bus.
Survey of software architecture- Round robin, round robin with interrupts, function queue scheduling architecture, selecting an architecture saving memory space.
Embedded software development tools- Host and target machines, linkers, locations for embedded software, getting embedded software into target system, debugging technique.
RTOS concepts - Architecture of the kernel, interrupt service routines, semaphores, message queues, pipes.
Instruction sets- Introduction, preliminaries, ARM processor, SHARC processor.
System design techniques - Design methodologies, requirement analysis, specifications, system analysis and architecture design.
Design examples- Telephone PBX, ink jet printer, water tank monitoring system, GPRS, Personal Digital Assistants, Set Top boxes.

References:

1. Computers as a component: principles of embedded computing system design- wayne wolf
2. An embedded software premier: David E. Simon
3. Embedded / real time systems-KVKK Prasad, Dreamtech press, 2005
4. Programming for embedded system by Dr. Prasas, Vikas Gupta, Das & Verma, Pub, WILEY Dreamtech india Pvt.
5. Embadded System Design. by Frank Vashid & Tony Givergis, Pub, WILEY.
6. MFC Programming. by Herbert Schildt, Pub. TataMcGraw Hill.
7. Programming Embedded Systems by Michael Barr, Pub. O'REILLY

Multimedia

Introduction Multimedia and its Application, Different Media, Hypertext and Hypermedia, Issues in Multimedia System, Component of a Multimedia System.
Overview of Text and Graphics: Types of Text Data (Plain/Formatted/Hypertext), Unicode Scheme, Concept of Font, File Formats (txt, doc, rtf, ps, pdf etc.), Vector and Raster Graphics.
Image: Image Digitization, Digital Image, Binary/GrayScale/ Colour Image, Colour Models, File Formats, Overview of Contrast Intensification, noise removal, edge detection and segmentation Image Descriptors (Shape, Texture and Colour Features).
Loss-less and Lossy Image Compression including JPEG.
An overview of Content Based Image Retrieval System.

[Audio: Audio Digitization (Sampling and Quantization, Representation based on PCM/DPCM/DM/ADM), File Formats.
Time Domain Descriptors (ZCR, STE etc.), Frequency Domain Descriptors (Spectral Centroid, Spectral Flux, Spectral Roll Off etc.), and Perception based Descriptors (Mel Scale, MFCC).
Psycho Acoustics and Audio Compression.

An Overview of Audio Classification/Retrieval System.
Video: Structure of Video Data, File Formats.
Video Compression. Motion Estimation. Structural Segmentation of Video Data. Overview of Video Summarization, Browsing and Retrieval System.
Animation: Keyframes & tweening, cel & path animation, principles and techniques of animation, Web animation, 3D animation principles, camera, special effects, transformations and editing, rendering algorithms, features of animation software, file formats.

References:

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods
2. Digital Image Processing and Analysis by B. Chanda and D. Dutta Majumder
3. Principles of Multimedia by Ranjan Parekh
4. Multimedia –A Practical Approach by Sanhker, Jaico.
5. Multimedia Systems by Buford J. K., Pearson Education.
6. Multimedia and Imaging Databases by S. Khoshafian, A. Brad Baker, Morgan Kaufmann.
7. Multimedia Systems Design, Prabhat k. Andleigh & Kiran Thakkar, Prentice Hall PTR.
8. Digital Multimedia by Nigel Chapman & Jenny Chapman, John-Wiley.

Computational Geometry

Geometric Objects – Points, Lines, Planes, Polygons, 3D Objects – Geometric Algorithms – Degeneracies and Robustness – Application Domains.

Convex Hull in 2D – Incremental Algorithm.

Line Segment Intersection Algorithms – Doubly Connected Edge List – Map Overlays – Boolean operations.

Polygon Triangulation – Partitioning Polygons into Monotone Pieces – Triangulation of Monotone Polygons – Art Gallery Problem.

Half Plane Intersections – Use of Linear Programming Techniques – Manufacturing with Moulds

Orthogonal Range Searching – Kd Trees – Range Trees – Higher Dimensional Range Trees

Database Searching – Point Location.

Voronoi Diagrams – VD of Line Segments – Farthest Point VDs – Post Office Problem 6L Convex Hulls in 3-space.

Robot Motion Planning – Work Space and Configuration Space – Translational Motion Planning

References:

1. Computational Geometry – Algorithms and Applications by Berg, Cheong, Kreveld and Overmars 3e, Springer
2. Computational Geometry – An Introduction by Preparata and Shamos, Springer
3. Computational Geometry in C – Joseph O'Rourke, 2e, Cambridge Univ Press

Cloud Computing

Introduction: Essentials, Benefits and need for Cloud Computing - Business and IT Perspective - Cloud and Virtualization - Cloud Services Requirements - Cloud and Dynamic Infrastructure - Cloud Computing Characteristics Cloud Adoption.
Cloud Models: Cloud Characteristics - Measured Service - Cloud Models - Security in a Public Cloud Public versus Private Clouds - Cloud Infrastructure Self Service.
Cloud as a Service: Gamut of Cloud Solutions - Principal Technologies - Cloud Strategy Cloud Design and Implementation using SOA - Conceptual Cloud Model - Cloud Service Defined

Cloud Solutions: Cloud Ecosystem - Cloud Business Process Management - Cloud Service Management - Cloud Stack - Computing on Demand (CoD) – Cloud sourcing.

Cloud Offerings: Information Storage, Retrieval, Archive and Protection - Cloud Analytics Testing under Cloud - Information Security - Virtual Desktop Infrastructure - Storage Cloud.

Cloud Management: Resiliency – Provisioning - Asset Management - Cloud Governance - High Availability and Disaster Recovery - Charging Models, Usage Reporting, Billing and Metering.

Cloud Virtualization Technology: Virtualization Defined - Virtualization Benefits - Server Virtualization - Virtualization for x86 Architecture - Hypervisor Management Software - Logical Partitioning (LPAR) - VIO Server - Virtual Infrastructure Requirements.

Cloud Virtualization: Storage virtualization - Storage Area Networks - Network-Attached storage - Cloud Server Virtualization - Virtualized Data Center.

Cloud and SOA: SOA Journey to Infrastructure - SOA and Cloud - SOA Defined - SOA and IaaS - SOA-based Cloud Infrastructure Steps - SOA Business and IT Services.

Cloud Infrastructure Benchmarking: OLTP Benchmark - Business Intelligence Benchmark - e-Business Benchmark - ISV Benchmarks - Cloud Performance Data Collection and Performance Monitoring Commands - Benchmark Tools.

References:

1. Cloud Computing – Insight into New Era Infrastructure, Dr. Kumar Saurabh, Wiley India.
2. Cloud Computing Explained, John Rhoton, Recursive Press
3. Cloud Computing Bible, Barry Sosinsky, Wiley
4. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg, Wiley
5. Cloud Computing for Dummies, Judith Hurwiz, Wiley Publishing.
6. The Cloud at your service, Rosenberg and Matheos, Manning Publications

Combinatorial Algorithms

1. Generation of Elementary Combinatorial Objects;

2. Exhaustive Search, i.e. Backtracking

3. Computing auto morphisms and determining isomorphism of combinatorial objects.

4. Heuristic search algorithms such as Hill Climbing, Tabu Search, Simulated Annealing, and

Genetic Algorithms.

Reference

1. D.L. Kreher and D.R. Stinson, *Combinatorial Algorithms: Generation, Enumeration and Search* CRC press, 1998.

COS-491: AI Lab

Course Outcomes :

On successful completion of this course, students will be able to

- Learn programming concept in PROLOG platform
- Improve logic building for programming

Artificial Intelligence Program using PROLOG.

List of Assignments:

1. Study of PROLOG.
2. Write the following programs using PROLOG:
3. Write a program to solve 8-queens problem.

4. Solve any problem using depth first search.
5. Solve any problem using best first search.
6. Solve 8- puzzle problem using best first search.
7. Solve Robot (traversal) problem using means End Analysis.
8. Solve Traveling Salesman problem.

COS-492: Project work

Course Outcomes :

On successful completion of this course, students will be able to

- understand how to work at corporate field.
- comprehend the state-of-the-art requirements of the Industry.
- apply critical thinking, reasoning and creative thinking for Software Design in an industry as an individual or as a part of a team.
- analyze the problem and provide Solution by Decision Making.
- develop Interpersonal, Communication and Presentation skills.
- build the modules for a specific problem.

A separate project will be assigned to each student under the supervision of internal faculty members. The students will prepare a project report in consultation with the supervisor allotted by the department committee which will be presented before a board of examiners to be nominated by the B.O.S.

COS-493 Grand Viva

Course Outcomes :

On successful completion of this course, students will be able to

- analyze the problem and provide Solution by Decision Making.
- develop Interpersonal, Communication and Presentation skills.
- build the modules for a specific problem.

3. Program Specific Outcomes (PSO)

The program is designed to fulfill the following specific outcomes.

- Enable the students to apply the computing and soft skills acquired in the
- M.Sc. in Computer Science program and developing innovative applications for the betterment of the society.
- Provide exposure to techniques that would enable the students to design, implement and evaluate IT solutions.
- To enable the students to meet the challenges of research and development in computer science and applications.