VIDYASAGAR UNIVERSITY

MIDNAPORE-721102

The SYLLABUS for

POST- GRADUATE Courses

in

COMPUTER SCIENCE

Under Choice Based Credit System (CBCS)

(Semester Programme)

[w. e. f. 2018-19 session]

Programme Outcomes

On completion of the M.Sc.(Computer science) students are able to:

- Serve as the Programmers or the Software Engineers with the sound knowledge of practical and theoretical concepts for developing software.
- Can motivate in their research work.
- Serve as the Computer Engineers with enhanced knowledge of computers and its building blocks.
- Work as the System Engineers and System integrators.
- Serve as the System Administrators with thorough knowledge of RDBMS.
- To Give Technical Support for the various systems.
- Work as the Support Engineers and the Technical Writers.
- Serve as System Analyst in different organization.
- Work as Consultant and Management officers for system management.
- Serve as the IT Officers in Banks and cooperative societies.
- Serve as the Web Designers with latest web development technologies.
- Serve as Teacher at Secondary School Level.
- Serve as Computer Scientist in research and R & D laboratories.
- Serve as Research Staff Member

FIRST YEAR FIRST SEMESTER

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 20	05 05	4
COS-203	M-I:OOPS with Java M-II: Programming in R	3	1	0	40	10	4
COS-204 (CBCS)	M-I: Computer Fundamentals M-II: Programming Concepts	2 2	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	40	10	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 M-II: Term Paper	0 0	0 0	0 3	0 0	25 25	3
		14	2	9	200	100	22

Total Period/Week = 25 Total Marks = 300

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:

<u>Elective 1, Code: COS-303</u>

(A) Graph Theory
(B) Distributed computing
(C) Mobile Computing
(D) Pattern Recognition
(E) Machine Learning
(F) Soft Computing
(G) Recent Computing

<u>Elective II, Code : COS-402</u>

(A) Web Technology
(B) Data Mining
(C) Parallel Computing
(D) Cryptography and Steganography
(E) Bio Informatics
(F) Natural Language Processing
(G) Data Science

<u>Elective III, Code : COS-403</u>

(A) Information Security
(B) Cognitive Computing
(C) Embedded System
(D) Multimedia
(E) Computational Geometry
(F) Combinatorial Algorithm
(G) Cloud Computing

M.Sc. (1st Semester)

COS-101: Data Structure and algorithm **Lectures: 40**

Objective: The main objective of data structure is to understand different data structures with data organization and different associated operations those are applied on data.
Detail Syllabus:

Course Outcome : After completion of the course, students will able to understand the following

- Basic properties of an ADT.
- Different data structures and associated operations

- Big O notation for time complexity
- Big O notation for time complexity
- Thoroughly knowledge about array and linked list
- Different non - linear data structure like tree and balanced trees.
- Different searching algorithms and time complexity

Details Syllabus

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 **Lectures: 40**

Objective: In the field of computer science, this course explains how computers are designed, build and operate, a popular performance improvement method of a computer system known as pipelining, super pipeline and super scalar design, hierarchical memory technology, performance of cache memory with its mapping and virtual memory concept, vector and array processors principles, multiprocessor architecture. It helps us design and implement applications much better after knowing what's inside a machine and how it works.

Course Outcome :

At the end of this course students should:

- Know instruction set architecture, ALU & CU units, memory and I/O devices, measuring performance, CISC and RISC processors.
- Understand basic concepts of pipelining and its different hazards with solutions.
- Understand hierarchical memory technology and its properties, cache memory, virtual memory.
- Understand instruction-level parallelism, superscalar, super-pipelined and VLIW processor.
- Understand vector and array processors principles, multiprocessor architecture.

Detail Syllabus:

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 Lectures: 40

Objective: The main emphasis of this section is on the organization and management of local area networks (LANs). The course objectives include learning about computer network organization and implementation, obtaining a theoretical understanding of data communication and computer networks, and gaining practical experience in installation, monitoring, and troubleshooting of current LAN systems. The course introduces computer communication network design and its operations. The course includes the following topics: Open Systems Interconnection (OSI) communication model; error detection and recovery; local area networks; bridges, routers and gateways; network naming and addressing; and local and remote procedures. On completion of the course, the student should be able in part to design, implement and maintain a typical computer network (LAN).

Outcome:

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

- Describe the general principles of data communication.
- Describe how computer networks are organized with the concept of layered approach.
- Describe how signals are used to transfer data between nodes.
- Implement a simple LAN with hubs, bridges and switches.
- Describe how packets in the Internet are delivered.
- Analyze the contents in a given Data Link layer packet, based on the layer concept.
- Design logical sub-address blocks with a given address block.
- Decide routing entries given a simple example of network topology
- Describe what classless addressing scheme is.
- Describe how routing protocols work.

Detail Syllabus:

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 **Lectures: 40**

Objective:

1. Knowledge of basic SW engineering methods and practices, and their appropriate application.
2. Describe software engineering layered technology and Process frame work.
3. A general understanding of software process models such as the waterfall and evolutionary models.
4. Understanding of software requirements and the SRS documents.
5. Understanding of the role of project management including planning, scheduling, risk management, etc.
6. Describe data models, object models, context models and behavioural models.
7. Understanding of different software architectural styles.
8. Understanding of implementation issues such as modularity and coding standards.
9. Understanding of approaches to verification and validation including static analysis, and reviews.
10. Understanding of software testing approaches such as unit testing and integration testing.
11. Describe software measurement and software risks.
12. Understanding of software evolution and related issues such as version management.
13. Understanding on quality control and how to ensure good quality software.

Outcome:

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.
3. Ability to develop, maintain and evaluate large-scale software systems.
4. To produce efficient, reliable, robust and cost-effective software solutions.
5. Ability to perform independent research and analysis.
6. To communicate and coordinate competently by listening, speaking, reading and writing english for technical and general purposes.
7. Ability to work as an effective member or leader of software engineering teams.
8. To manage time, processes and resources effectively by prioritising competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.
9. Ability to understand and meet ethical standards and legal responsibilities.

Detail Syllabus:

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 **Lectures: 60**

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 **Lectures: 60**

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)

M.Sc. (2nd Semester)

COS-201: Advanced Database Management System **Lectures: 40**

Objective: This course is intended to provide you with an understanding of the current theory and practice of database management systems. To help you more fully appreciate their nature, the course provides a solid technical overview of database management systems, using a current database product as a case study. In addition to technical concerns, more general issues are emphasized. These include data independence, integrity, security, recovery, performance, database design principles, and database administration.

Outcome:

At the completion of this course, students should be able to do thefollowing:

1. Understand the role of a database management system in an organization.

2. Understand basic database concepts, including the structure and operation of the relational data model.

3. Construct simple and moderately advanced database queries using Structured Query Language (SQL).

4. Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.

5. Design and implement a small database project using SQL.

6. Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.

7. Describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.

8. Understand the role of the database administrator.

Detail Syllabus:

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 — Lectures: 20

Objective: Course should provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (and less magical) view towards algorithmic design and in general computation itself. The course should in addition clarify the practical view towards the applications of these ideas in the theoretical computer science.

Outcome: After completing the course, the student will be able to:

1.Model, compare and analyse different computational models using combinatorial methods.

2. Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.

3. Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.

4. Identify limitations of some computational models and possible methods of proving them.

5. Have an overview of how the theoretical study in this course is applicable to and engineering application like designing the compilers.

Detail Syllabus:

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.

COS-202:M2: Compiler Construction Lectures: 20

Objective: The main objective of this course is to introduce the major concept areas of language translation and compiler design and to develop an awareness of the function and complexity of modern compilers. This course is a study of the theory and practice required for the design and implementation of interpreters and compilers for programming languages.

Outcome:

By the end of the course, the successful student will be able to do:

- To realize basics of compiler design and apply for real time applications.
- To introduce different translation languages
- To understand the importance of code optimization
- To know about compiler generation tools and techniques
- To learn working of compiler and non-compiler applications
- Design a compiler for a simple programming language
- To convert from source language to target language and should be recognize what happens at each and every phase of a compiler.
- To understand the different types of parsing techniques and should be in a position to solve the problem.
- To write the code by using YACC and lex.

Detail Syllabus:

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).
2. 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.
4. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2003.
5. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
6. Kenneth C. Louden, "Compiler Construction: Principles and Practice", Thomson Learning.
7. Systems Programming and Operating Systems – D. M. Dhamdhere, TMH
8. John J. Donovan, " Systems Programming", 3rd edition, 1997, Addison Wesley.

COS-203:M1: OOPS using JAVA Lectures: 20

Objective: Java is the most popular platform, which is used to develop several applications for the systems as well as embedded devices like mobile, laptops, tablets and many more. It is an object oriented programming language and has a simple object model, as it has derived from C and C++. It provides a virtual machine, which is accumulated with byte-code and can run on any system. With time the importance and popularity of Java is on rise as it has the magic in its remarkable abilities to innovate and morph as the technology landscape changes. It is the language of choice for developing applications for the BlackBerry Smartphone. It is important for information technology industry to develop and create multiple web-based or server based applications to enhance the industrial competency. There is huge scope for this programming language.

Outcome:

After completing this course the student must demonstrate the knowledge and ability to:
Able to understand the use of OOPs concepts.
Able to solve real world problems using OOP techniques.
Able to understand the use of abstraction.
Able to understand the use of Packages and Interface in java.

Able to develop and understand exception handling, multithreaded applications with synchronization.
Able to understand the use of Collection Framework.
Able to design GUI based applications and develop applets for web applications.

Detail Syllabus:

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

COS-203:M2: Programming in R Lectures: 20

Objective: The main goal of this course is to help students learn, understand, and practice big data analytics and machine learning approaches, which include the study of modern computing big data technologies and scaling up machine learning techniques focusing on industry applications. Mainly the course objectives are: conceptualization and summarization of big data and machine learning, trivial data versus big data, big data computing technologies, machine learning techniques, and scaling up machine learning approaches.

Outcome:

The students learning outcomes are designed to specify what the students will be able to perform after completion of the course:
• Ability to identify the characteristics of datasets and compare the trivial data and big data for various applications.
• Ability to select and implement machine learning techniques and computing environment that are suitable for the applications under consideration. .
• Ability to solve problems associated with batch learning and online learning, and the big data characteristics such as high dimensionality, dynamically growing data and in particular scalability issues.
• Ability to understand and apply scaling up machine learning techniques and associated computing techniques and technologies.
• Ability to recognize and implement various ways of selecting suitable model parameters for different machine learning techniques.

• Ability to integrate machine learning libraries and mathematical and statistical tools with modern technologies like hadoop and mapreduce.

Detail Syllabus:

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 (CBCS) Lectures: 20

Objective :

The objectives of this course are

1. This course introduces the concepts of computer basics & programming with particular attention to Engineering examples.

2. The C programming language is used but the course will stress on fundamental parts of programming language, so that the students will have a basic concept for understanding and using other programming language.

Outcome:

On completion of the course students will be able to

1. Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming

2. Write, compile and debug programs in C language and use different data types for writing theprograms.

3. Design programs connecting decision structures, loops and functions.

4. Explain the difference between call by value and call by address.

5. Understand the dynamic behavior of memory by the use of pointers.

Detail Syllabus:

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.

COS-204: M2: Programming Concepts (CBCS) Lectures: 20

Objective:

1. Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.

2. Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.

3. Be aware of the important topics and principles of software development.

4. Have the ability to write a computer program to solve specified problems.

5. Be able to use the Java SDK environment to create, debug and run simple Java programs.

Outcome:

After successful completion of the course student can write and execute computer program in C.

Detail Syllabus:

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 Lectures: 60

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 **Lectures: 30**

1. Assignments on class, constructor, overloading, inheritance, overriding
2. Assignments on wrapper class, vectors, arrays
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

COS-291: M2: R Lab **Lectures: 30**

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

M.Sc. (3^{rd} Semester)

COS-301: Advanced operating System **Lectures: 40**

Objective:

1. To learn the fundamentals of Operating Systems.

2. To learn the mechanisms of OS to handle processes and threads and their communication

3. To learn the mechanisms involved in memory management in contemporary OS

4. To gain knowledge on distributed operating system concepts that includes architecture,

Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols

5. To know the components and management aspects of concurrency management

6. To learn programmatically to implement simple OS mechanisms

Outcome:

Students will be able to:

1: Analyze the structure of OS and basic architectural components involved in OS design

2: Analyze and design the applications to run in parallel either using process or thread models of different OS

3: Analyze the various device and resource management techniques for timesharing and distributed systems

4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system

5: Interpret the mechanisms adopted for file sharing in distributed Applications

6: Conceptualize the components involved in designing a contemporary OS

Detail Syllabus:

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 Lectures: 20

Objective:

Students will try to learn:
1. To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
2. To learn the basic principles of 3- dimensional computer graphics.
3. Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.

4. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.
5. To be able to discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications.

Outcome:

Students will able to:

1. To list the basic concepts used in computer graphics.

2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.

3. To describe the importance of viewing and projections.

Detail Syllabus:

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

COS-302:M2: Image Processing Lectures: 20

Objective:

Students will able to:

1. Fundamental concepts of a digital image processing system. 2. Concepts of image enhancement techniques. 3. Various Image Transforms. 4. Compression techniques and Morphological concepts 5. Various segmentation techniques, and object descriptors. 6. Color models and various applications of image processing.

Outcome:

Students will able to:

1. Remember the fundamental concepts of image processing. 2. Explain different Image enhancement techniques 3. Understand and review image transforms 4. Analyze the basic algorithms used for image processing &image compression with morphological image processing. 5. Contrast Image Segmentation and Representation 6. Design & Synthesize Color image processing and its real world applications.

Detail Syllabus:

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

Image Enhancement: Contrast stretching, Histogram Equalization, Binarization [4L] 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 Lectures: 40

Graph Theory

Objective:

1. To understand and apply the fundamental concepts in graph theory

2. To apply graph theory based tools in solving practical problems

3. To improve the proof writing skills.

Outcome:

- The students will be able to apply principles and concepts of graph theory in practical situations

Detail Syllabus:

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

Objective:

This course provides an introduction to the fundamentals of distributed computer systems, assuming the availability of facilities for data transmission.

Outcome:

• Study software components of distributed computing systems. Know about the communication and interconnection architecture of multiple computer systems.

• Recognize the inherent difficulties that arise due to distributed-ness of computing esources. Understanding of networks & protocols, mobile & wireless computing and their applications to real world problems.

• At the end students will be familiar with the design, implementation and security issues of distributed system.

Detail Syllabus:

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

Objective:

This course will provide graduate students of MSc Information Systems with both broad and in-depth knowledge, and a critical understanding of mobile computing from different viewpoints: infrastructures, principles and theories, technologies, and applications in different domains. The course will provide a complete overview of the mobile computing subject area, including the latest research.

Outcome:

Upon successful completion of this course, you will be able to

- explain the principles and theories of mobile computing technologies.
- describe infrastructures and technologies of mobile computing technologies.
- list applications in different domains that mobile computing offers to the public, employees, and businesses.
- describe the possible future of mobile computing technologies and applications.
- effectively communicate course work through written and oral presentations.

Detail Syllabus:

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

Objective:

The course is designed to introduce students to theoretical concepts and practical issues associated with pattern recognition

Outcome:

1. Design systems and algorithms for pattern recognition (signal classification), with focus on sequences of patterns that are analyzed using, e.g., hidden Markov models (HMM),

2. Analyse classification problems probabilistically and estimate classifier performance,

3. Understand and analyse methods for automatic training of classification systems,

4. Apply Maximum-likelihood parameter estimation in relatively complex probabilistic models, such as mixture density models and hidden Markov models,

5. Understand the principles of Bayesian parameter estimation and apply them in relatively simple probabilistic models

Detail Syllabus:

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, MST-basedmethod, 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

Objective:

This course will serve as a comprehensive introduction to various topics in machine learning. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems; and be able to evaluate and interpret the results of the algorithms.

Outcome:

By the end of the module, students should:

- develop an appreciation for what is involved in learning from data.
- understand a wide variety of learning algorithms.
- understand how to apply a variety of learning algorithms to data.
- understand how to perform evaluation of learning algorithms and model selection.

Detail Syllabus:

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

Objective:

The primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of soft computing.

Upon successful completion of the course, students will have an understanding of the basic areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.

Provide the mathematical background for carrying out the optimization associated with neural network learning.

Aim of this course is to develop some familiarity with current research problems and research methods in Soft Computing by working on a research or design project.

Outcome:

The student will be able to:

Describe human intelligence and AI

Explain how intelligent system works.

Apply basics of Fuzzy logic and neural networks.

Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience

Relate with neural networks that can learn from available examples and generalize to form appropriate rules for inference systems

Describe with genetic algorithms and other random search procedures useful while seeking global optimum in self-learning situations

Develop some familiarity with current research problems and research methods in Soft Computing Techniques.

Detail Syllabus:

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

Objective:

1. To learn how to use Distributed computing architechture.

2. To learn grid computing

3. To learn parallel computing.

4, To learn cloud computing.

Outcome:

Able to know what are the advance computing techniques nowadays.

Detail Syllabus:

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 (CBCS) Lectures: 20

Objective:

The objective of the course is to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS.

Course Outcome : Upon successful completion of this course, students should be able to:

• Describe the fundamental elements of relational database management systems

• Explain the basic concepts of relational data model, entity-relationship model, relationaldatabase design, relational algebra and SQL.

• Design ER-models to represent simple database application scenarios

• Convert the ER-model to relational tables, populate relational database and formulate SQLqueries on data.

• Improve the database design by normalization.

• Familiar with basic database storage structures and access techniques: file and page

organizations, indexing methods including B tree, and hashing.

Detail Syllabus:

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

COS-304: M2: Internet Technology (CBCS) Lectures: 20

Detail Syllabus:

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 **Lectures: 30**

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.

COS-391: M2: OS Lab Lectures: 30

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 Lectures: --

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.

COS-391: M1: Term Paper Lectures: 30

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

M.Sc. (4^{th} Semester)

COS-401: Artificial Intelligence Lectures: 40

Objective:

Students will try to learn:

1. To create appreciation and understanding of both the achievements of AI and the theory underlying those achievements.

2. To introduce the concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems

3. To review the different stages of development of the AI field from human like behavior to Rational Agents.

4. To impart basic proficiency in representing difficult real life problems in a state space representation so as to solve them using AI techniques like searching and game playing.

5. To create an understanding of the basic issues of knowledge representation and Logic and blind and heuristic search, as well as an understanding of other topics such as minimal, resolution, etc. that play an important role in AI programs.

6. To introduce advanced topics of AI such as planning, Bayes networks, natural language processing and Cognitive Computing

Outcome:

Students will able to:

1. Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.

2. Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.

3. Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing

4. Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.

5. Formulate and solve problems with uncertain information using Bayesian approaches.

6. Apply concept Natural Language processing to problems leading to understanding of cognitive computing.

Detail Syllabus:

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 **Lectures: 40**

Web Technology

Objective:

The objective of this lab is to To develop an ability to design and implement static and dynamic website.

Outcome:

On completion of this course, a student will be familiar with client server architecture and able to develop a web application using java technologies. Students will gain the skills and project-based experience needed for entry into web application and development careers.

Detail Syllabus:

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

<u>**Data Mining**</u>

Objective:

Students undergoing this course are expected to:

• Differentiate OnLine Transaction Processing and OnLine Analytical processing

• Learn Multidimensional schemas suitable for data warehousing

• Understand various data mining functionalities

• Inculcate knowledge on data mining query languages.

• Know in detail about data mining algorithms

Outcome:

After undergoing the course, Students will be able to understand

• Design a data mart or data warehouse for any organization

• Develop skills to write queries using DMQL

• Extract knowledge using data mining techniques

• Adapt to new data mining tools.

• Explore recent trends in data mining such as web mining, spatial-temporal mining

Detail Syllabus:

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

Objective:

Students will demonstrate an understanding of concepts, algorithms, and design principles underlying parallel computing, develop algorithm design and implementation skills, and gain practical experience in programming large scale parallel machines

Outcome:

The course outcomes are skills and abilities students should have acquired by the end of the course.

- Students demonstrate they can define and apply parallel computing to a variety of applications in Mathematics and Engineering.

- Students will have an ability to assess a problem presented to them, design a solution, and test their implementation.
- Students will be presented with problems and will have to design and implement solutions for those problems
- Students will have an ability to discuss large scale machine design as well as applications and algorithms on those machines.

Detail Syllabus:

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

Objective:

1. To understand basics of Cryptography and Steganography.
2. To be able to secure a message over insecure channel by various means.

3. To learn about how to maintain the Confidentiality, Integrity and Availability of a data.

Outcome:

After successful completion of the course, the learners would be able to

1. Provide security of the data over the network.

2. Do research in the emerging areas of cryptography and steganography.

3. Implement various networking protocols.

4. Protect any network from the threats in the world.

Detail Syllabus:

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

Objective:

Students undergoing this course are expected to:

• Understand the basic concepts and techniques of Bioinformatics.

• Develop an awareness of the computational problems that arise in the modeling and analysis of living

systems.

• Understand basic abstractions and computational approaches used to formulate and address

these problems.

Outcome:

After undergoing the course, Students will be able to understand

• Sequencing Alignment and Dynamic Programming

• Sequence Databases

Detail Syllabus:

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

Objective:

To understand natural language processing and to learn how to apply basic algorithms in this field. To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

Outcome:

The students will get acquainted with natural language processing and learn how to apply basic algorithms in this field. They will understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. They will also grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

Detail Syllabus:

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

Data Science

COS-403: Elective – III Lectures: 40

Information Security

Objective:

Students are expected to demonstrate the ability to:

1. Identify computer and network security threats, classify the threats and develop a security model to prevent, detect and recover from the attacks. (ABET Outcomes: a, c, e, j, k)

2. Encrypt and decrypt messages using block ciphers, sign and verify messages using well known signature generation and verification algorithms. (ABET Outcomes: c, e, k)

3. Analyze existing authentication and key agreement protocols, identify the weaknesses of these protocols. (ABET Outcomes: c, e, k)

4. Download and install an e-mail and file security software, PGP, and efficiently use the code to encrypt and sign messages. (ABET Outcomes: c, e, k)

5. Develop SSL or Firewall based solutions against security threats, employ access control techniques to the existing computer platforms such as Unix and Windows NT. (ABET Outcomes: a, c, e, i, k)

Detail Syllabus:

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

Objective:

1. To have knowledge about the basic working of a microcontroller system and its programming in assembly language.

2. To provide experience to integrate hardware and software for microcontroller applications systems.

Outcome:

To acquire knowledge about microcontrollers embedded processors and their applications.

1. Foster ability to understand the internal architecture and interfacing of different

peripheral devices with Microcontrollers.

2. Foster ability to write the programs for microcontroller.

3. Foster ability to understand the role of embedded systems in industry.

4. Foster ability to understand the design concept of embedded systems.

Detail Syllabus:

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

<u>Multimedia</u>

Objective:

Students will try to learn:

1. To learn and understand technical aspect of Multimedia Systems.

2. To understand the standards available for different audio, video and text applications.

3. To Design and develop various Multimedia Systems applicable in real time.

4. To learn various multimedia authoring systems.

5. To understand various networking aspects used for multimedia applications.

6. To develop multimedia application and analyze the performance of the same

Outcome:

Students will able to:

1. Developed understanding of technical aspect of Multimedia Systems.

2. Understand various file formats for audio, video and text media.

3. Develop various Multimedia Systems applicable in real time.

4. Design interactive multimedia software.

5. Apply various networking protocols for multimedia applications.

6. To evaluate multimedia application for its optimum performance.

Detail Syllabus:

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

Objective:

The objectives of this course are as follows:

(a) Introduce rigorous algorithmic analysis for problems in Computational Geometry.

(b) Discuss applications of Computational Geometry to graphical rendering.

(c) Introduce the notions of Voronoi diagrams and Delaunay Triangulations.

(d) Develop expected case analyses for linear programming problems in small dimensions.

Outcome:

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

(a) Analyze randomized algorithms for small domain problems.

(b) Use line-point duality to develop efficient algorithms.

(c) Apply geometric techniques to real-world problems in graphics.

(d) Solve linear programs geometrically

Detail Syllabus:

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

Objective:

1. To learn how to use Cloud Services.

2. To implement Virtualization

3. To implement Task Scheduling algorithms.

4. Apply Map-Reduce concept to applications.

5. To build Private Cloud.

6. Broadly educate to know the impact of engineering on legal and societal issues involved.

Outcome:

The primary learning outcomes of this course are five-fold. Students will be able to:

1) Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.

2) Apply fundamental concepts in cloud infrastructures to understand the tradeoffs in power, efficiency and cost, and then study how to leverage and manage single and multiple datacenters to build and deploy cloud applications that are resilient, elastic and cost-efficient.

3) Discuss system, network and storage virtualization and outline their role in enabling the cloud computing system model.

4) Illustrate the fundamental concepts of cloud storage and demonstrate their use in storage systems such as Amazon S3 and HDFS.

5) Analyze various cloud programming models and apply them to solve problems on the cloud.

Detail Syllabus:

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

COS-491: AI Lab **Lectures: 30**

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 **Lectures: --**

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-391: M1: Graphics Lab **Lectures: 30**

- Point plotting, line & regular figure algorithms
- Raster scan line, circle and ellipse drawing algorithms
- Clipping algorithms for points, lines & polygons
- 2-D transformations
- Filling algorithms.
- Curve drawing