Vidyasagar University
Midnapore-721102, West Bengal

The SYLLABUS for
POST-GRADUATE Programme

in

COMPUTER SCIENCE

Under Choice Based Credit System (CBCS)
(Semester Programme)

[w.e.f. 2022-23]

Programme Outcomes

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

- Get core competence in various subjects of Computer Science
- Recognize the organizational need and to engage themselves in continuing professional development.
- Design, implement, and evaluate a computational system to meet the desired needs within realistic constraints.
- Can motivate them in their research work.
- Recognize the need for and ability to engage in continuing professional development.
- Identify, formulate, develop solutions to computational challenges. Understand professional, ethical, legal, security, and social issues and responsibilities for the computing profession.

Programme Specific Outcomes

- Able to handle any kind of software development
- Able to maintain the software network to handle the technological challenges.
- Able to develop strong analytical skills, critical thinking and experimental skills.
- Able to solving on Computational problems, system networking knowledge, use of technology with innovative ideas

Distinctive feature of the courses:

- **Value-added course:** COS 304.2,
- **Employability/entrepreneurship/ skill development:** COS 201.1, 204.2, 301.2.
- **Digital content:** COS 196.2, 296.2
- **Ethics, gender, human values, environment & sustainability:** COS 201.2
- **The new course introduced:** COS 202.1, 202.2, 302.1, 302.2, 303.E, 303.F

FIRST YEAR FIRST SEMESTER

Subjects		Marks Distribution		Credit Points
Course Code	Course Name	Theoretical / Practical Examination	Internal / Sessional	
COS-101	Analysis of Algorithms	40	10	4 (3-1-0)
COS-102	M1: Parallel Computing M2: Distributed System	20 20	05 05	4 (3-1-0)
COS-103	M1: Pattern Recognition M2: Image Processing	20 20	05 05	4 (3-1-0)
COS-104	Real Time System	40	10	4 (3-1-0)
COS-195	M1: Parallel Computing Lab M2: Image Processing Lab	20 20	05 05	4 (0-0-8)
COS-196	M1: Algorithm Lab using Python M2: Cloud Computing Lab	40	10	4 (0-0-8)
		240	60	24

Total Period/Week = 28 Total Marks = 300

FIRST YEAR SECOND SEMESTER

Subjects		Marks Distribution		Credit Points
Course Code	Course Name	Theoretical / Practical Examination	Internal / Sessional	
COS - 201	M1: Advanced DBMS M2: Green Computing	20 20	05 05	4 (3-1-0)
COS- 202	M1: Theory of Computer Science M2: Compiler Design	20 20	05 05	4 (3-1-0)
COS- 203	M1: Artificial Intelligent M2: Soft Computing	40	10	4 (3-1-0)
COS- 204 (CBCS)	M1: Computer Fundamentals M2: Programming Concepts	20 20	05 05	4 (3-1-0)
COS- 295	M1: DBMS Lab M2: UML Lab	20 20	05 05	4 (0-0-8)
COS- 296	M1: AI Lab M2: Soft Computing Lab	40	10	4 (0-0-8)
		240	60	24

Total Period/Week = 28 Total Marks = 300

Add-on / Value added course:

1. Agile Software Development [Mysql, Java, PHP]
2. R Programming and Analytics

Audit course:

1. Emerging Technologies for environment, ecology and human development [SDG]

SECOND YEAR FIRST SEMESTER

Subjects		Marks Distribution		Credit Points
Course Code	Course Name	Theoretical / Practical Examination	Internal / Sessional	
COS-301	M1: Advanced Networking M2: Network Security	40	10	4 (3-1-0)
COS-302	M1: Machine Learning M2: Deep Learning	40	10	4 (3-1-0)
COS-303	Elective	40	10	4 (3-1-0)
COS-304 (CBCS)	M1: DBMS M2: Internet Technology	20 20	05 05	4 (3-1-0)
COS-395	M1: Advanced Networking Lab M2: Machine Learning Lab	20 20	05 05	4 (0-0-8)
COS-396	M1: Industrial Tour M2: Term Paper	0 0	25 25	4 (0-0-8)
		200	100	24

Total Period/Week = 25 **Total Marks = 300**

SECOND YEAR SECOND SEMESTER

Subjects		Marks Distribution		Credit Points
Course Code	Course Name	Theoretical / Practical Examination	Internal / Sessional	
COS-491	Project Work	0	200	12
COS-492	Grand Viva	0	50	6
COS-493	Term Paper	0	50	6
		0	300	24

Total Period/Week = 15 **Total Marks = 300**

Total Marks 1200; Total Credit 96

List of electives:

Elective Paper, Code: COS-303

A. Mobile Computing
B. Big Data Analytics
C. Cryptography and Steganography
D. Bioinformatics
E. IoT and Embedded System
F. Data Science
G. Quantum Computing
H. Natural Language Processing

MOOC / SWAYAM Course

(The list of courses offered in each semester will be provided at the beginning of the semester)

FIRST YEAR FIRST SEMESTER

M.Sc. (1st Semester)

COS-101: Analysis of Algorithm **Lectures: 40**

Course Objectives:

The main Course Objectives of algorithm and cryptography is to understand different algorithms and different associated operations that are applied on them.

Course Outcomes:

Upon completion of this course the students will argue the correctness of algorithms using inductive proofs and invariants. Analyze worst-case running times of algorithms using asymptotic analysis. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Synthesize dynamic-programming algorithms, and analyze them. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Synthesize greedy algorithms, and analyze them. Explain the major graph algorithms and their analyses. Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs. At the end they will learn about different cryptographic algorithms.

Detailed Syllabus:

Asymptotic Notation
Big-O, omega, theta etc.; finding time complexity of well known algorithms like- heapsort, search algorithm etc.

Recursion
Definition, Use, Limitations, Examples: Hanoi problem. Tail Recursion

Divide and Conquer
Basic method, Use, Examples: Merge sort, Quick Sort, Binary Search

Dynamic Programming
Basic method, use, Examples: matrix-chain multiplication, All pair shortest paths, single-source shortest path, Traveling Salesman problem

Branch and Bound
Basic method, use, Examples: The 15-puzzle problem

Backtracking
Basic method, use, Examples: Eight queens problem, Graph coloring problem, Hamiltonian problem

Greedy Method
Basic method, use, Examples: Knapsack problem, Job sequencing with deadlines

Lower Bound Theory
Bounds on sorting and sorting techniques using partial and total orders.

Disjoint Set Manipulation
Set manipulation algorithms like UNION-FIND, union by rank, Path compression.

Graph algorithms
BFS and DFS
Notion of NP-completeness
P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem.
Approximation algorithms
Necessity of approximation scheme, performance guarantee, Polynomial time approximation schemes: 0/1 knapsack problem.
Network Algorithms
Dijkstra's Algorithm, Minimum spanning tree algorithms (Prim's and Kruskal's algorithms)
Cryptography algorithms

DES, AES algorithm, RSA Algorithm, Diffie-Hellman algorithm, Secure Hash Algorithms.

Books:

1. Introduction to Algorithms by Cormen, Leiserson, Rivest& Stein, MIT Press
2. Fundamentals of computer algorithms by Horowitz, Sahni&Rajasekaran, Universities Press
3. Data Structues and Algorithms by Aho, Hopcroft& Ullman, Pearson
4. Cryptography and Network Security - Principles and Practice by William Stallings, Pearson
5. Cryptography And Network Security by Forouzan, McGraw Hill Education

COS-102: Parallel Computing and Distributed System **Lectures: 40**

Course Objectives:

The Course Objectives of this course is to provide knowledge in parallel computing, its organization and different parallel algorithmic techniques. It also provides information regarding distributed systems and cloud computing with its different deployment and service models.

Course Outcomes:

After completion of this course students will be able to use their knowledge of parallel computing and cloud computing in different applications as well as research fields.

Detailed Syllabus:
M1 : Parallel Computing **[20L]**
Introduction
From serial to parallel thinking, Performance metrics - speedup, utilization, efficiency, scalability, Models of Parallel Computation SIMD, MIMD, PRAM (EREW, CREW, CRCW), NC, How useful are these models for modern machines, Parallel Computer Organization, Pipelining and Throughput, Latency and Latency hiding, Memory Organization Inter-process communication, Inter connection network, Message passing, Shared/Distributed memory.
Basic Parallel Algorithmic Techniques
Pointer Jumping, Divide-and-Conquer, Partitioning, Pipelining, Accelerated Cascading, Synchronization (Locked, Lock-free)
Parallel Algorithms

Data organization for shared/distributed memory, Min/Max, Sum, Searching, Merging, Sorting, Prefix operations, Writing Parallel Programs, GPU- CUDA, Memory organization in CUDA, Multi-Core CPU programming, OpenMP, MPI, Performance evaluation and scalability

Books :

1. Introduction to Parallel Computing, AnanthGrama, 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

M2: Distributed System [20L]

Introduction to Clouds, Virtualization and Virtual Machine [3L]

Introduction to Distributed system and Cloud Computing, Utility computing, Features of today's Clouds, AAS Classification: HaaS, IaaS, PaaS, SaaS, Data-intensive Computing, New Cloud Paradigms, Categories of Clouds: Private clouds, Public clouds

Virtualization, Benefits of Virtualization, Virtualization Models: Bare metal, Hosted hypervisor, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para-virtualization, Device virtualization, Hot spot Mitigation for Virtual Machine Migration

Network Virtualization and Geo-distributed Clouds [3L]

Server Virtualization,SDN, Geo-distributed Cloud Data Centers: Inter-Data Center Networking, Data Center interconnection techniques: MPLS, Google's B4, and Microsoft's Swan

Leader Election in Cloud, Distributed Systems, and Industry Systems [3L]

Leader Election in Rings (Classical Distributed Algorithms & Ring LE & Bully LE Algorithm), Design of Zookeeper

Classical Distributed Algorithms and the Industry Systems [3L]

Time and Clock Synchronization in Cloud Data Centers, Global State and Snapshot Recording Algorithms, Distributed Mutual Exclusion

Consensus, Paxos and Recovery in Clouds [2L]

Consensus in a synchronous and asynchronous system, Paxos Algorithm, Byzantine Agreement, Failures & Recovery Approaches in Distributed Systems, Checkpointing and Recovery Algorithms.

Cloud Storage: Key-value stores/NoSQL [2L]

Design of Key-Value Stores, Design of HBase

P2P Systems and their use in Industry Systems [2L]

Peer to Peer Systems in Cloud Computing: Napster, Gnutella, FastTrack, BitTorrent, DHT, Chord, Pastry, and Kelips.

Cloud Applications: [2L]

MapReduce, Spark, and Apache Kafka

Books:

1. Distributed and Cloud Computing From Parallel Processing to the Internet of Things- Kai Hwang, Jack Dongarra, Geoffrey Fox.
2. Cloud Computing: Principles and Paradigms, Editors: RajkumarBuyya, James Broberg, Andrzej M. Goscinski, Wile, 2011

3. Distributed Computing: Principles, Algorithms, and Systems- Ajay D. Kshemkalyani and MukeshSinghal.
4. Distributed Computing: Fundamentals, Simulations and Advanced Topics-HagitAttiya and Jennifer Welch.

COS-103: Pattern Recognition and Image Processing

M1: Pattern Recognition **Lectures: 40**

Course Objectives:

- Explain and compare a variety of pattern classification, structural pattern recognition, and pattern classifier combination techniques.
- Apply performance evaluation methods for pattern recognition, and critique comparisons of techniques made in the research literature.
- Apply pattern recognition techniques to real-world problems such as document analysis and recognition.
- Implement simple pattern classifiers, classifier combinations, and structural pattern

Course Outcomes:

- Review the fundamental concepts of a pattern recognition system.
- Summarize, analyze, and relate research in the pattern recognition area verbally and in writing.
- Interpret simple pattern classifiers, classifier combinations, and structural pattern

Detailed Syllabus

Basic concepts of Pattern Recognition	**[1L]**
Pattern Preprocessing and Feature Selection	**[2L]**
Decision Functions	**[2L]**
Bayesian decision theory	**[2L]**
Parametric Estimation: Maximum likelihood estimation and Bayesian estimation	**[3L]**
Non- parametric Estimation: Parzen windows, Nearest Neighbor estimation	**[2L]**
Pattern Classification:	
Linear classifier: Perceptron, SVM	[**2L]**
Non-linear classifiers: MLP, Non-linear SVM	[**3L]**
Unsupervised learning and Clustering:	
Partitioning method, Density-based method, MST- basedmethod,	
Self organizing map, Hierarchical Clustering, Cluster validity	**[4L]**
Syntactic Pattern Recognition (Basic concepts)	**[2L]**
Some real-life applications	**[2L]**

Books:

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

M2: Image Processing

Course Objectives:

- Fundamental concepts of a digital image processing system.
- Concepts of image enhancement techniques.
- Various Image Transforms.
- Compression techniques and Morphological concepts
- Various segmentation techniques, and object descriptors.
- Color models and various applications of image processing.

Course Outcomes:

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

Detailed Syllabus:

Introduction
Fundamentals of Digital Image Processing, Image representation, Basic Image transforms, image file format .
Image Enhancement
Contrast stretching, Histogram Equalization, Binarization, Filtering in Spatial domain: Mean filter, Order Statistics filters.
Filtering in Frequency domain
Butterworth filter, Gaussian filter.
Image Restoration
Image degradation models, Weiner filter.
Image textures
Run Length Coding, Gray-level co-occurrence matrix
Image Segmentation
Edge detection: Gradient operators, Compass operator, Laplacian operators. LoG operator.
Region Segmentation
Region growing, region splitting and merging.
Shape detection
Least Mean Square error line fitting, Eigenvector line fitting, Straight line Hough Transform, Generalized Hough Transform.
Morphological Operators
Dilation, Erosion, Opening , Closing, Hit-and-Miss transforms, Applications. Image Compression.

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

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-104: Real time System **Lectures: 40**

Course Objectives:

- Understand the basics of the numerous applications of real-time systems.
- Acquire a wide understanding of the technologies and applications for real-time systems, a new and interesting subject.
- Obtain extensive hands-on experience in the design and development of a real operating system.

Course Outcomes:
On completion of this course, the students will be able to

- Acquiring a basic understanding of real-time systems and models
- Identifying the features of a real-time system
- Comprehend and develop a document on a real-time system's architectural design
- Create and document task scheduling, resource management, real-time operating systems, and fault-tolerant Real-Time System applications.

Detailed Syllabus:
Introduction [5L]
Definition, Typical Real Time Applications: Digital Control, High Level Controls, Signal Processing etc., Release Times, Dead-lines, and Timing Constraints, Hard Real Time Systems and Soft Real Time Systems, Reference Models for Real Time Systems: Processors and Resources, Temporal Parameters of Real Time Workload, Periodic Task Model, Precedence Constraints and Data Dependency.
Real Time Scheduling [7L]
Common Approaches to Real Time Scheduling: Clock Driven Approach, Weighted Round Robin Approach, Priority Driven Approach, Dynamic Versus Static Systems, Optimality of Effective-Deadline-First (EDF) and Least-Slack-Time-First (LST) Algorithms, Rate Monotonic Algorithm, Offline Versus Online Scheduling, Scheduling Aperiodic and Sporadic jobs in Priority Driven and Clock Driven Systems.
Resources Sharing [6L]
Effect of Resource Contention and Resource Access Control (RAC), Non-preemptive Critical Sections, Basic Priority-Inheritance and Priority-Ceiling Protocols, Stack Based Priority- Ceiling Protocol, Use of Priority-Ceiling Protocol in Dynamic Priority Systems, Preemption Ceiling Protocol, Access Control in Multiple-Module Resources, Controlling Concurrent Accesses to Data Objects.
Real Time Communication [8L]

Basic Concepts in Real time Communication, Soft and Hard RT Communication systems, Model of Real Time Communication, Priority-Based Service and Weighted Round-Robin Service Disciplines for Switched Networks, Medium Access Control Protocols for Broadcast Networks, Internet and Resource Reservation Protocols.

Real Time Operating Systems and Databases [7L]

Features of RTOS, Time Services, UNIX as RTOS, POSIX Issues, Characteristic of Temporal data, Temporal Consistency, Con-currency Control, Overview of Commercial Real Time databases.

Edge Computing [7L]

Introduction to Edge Computing, Origins of Edge, Edge Helping Low-End IoT Nodes, Architecture, Edge Helping Higher-Capability Mobile Devices: Mobile Offloading, Edge Helping the Cloud, Edge for Augmented Reality, Data Processing on the Edge, Dispersed Learning with Edge Computing, Video Analytics on the Edge, Edge Computing Real Time Application.

Text Books / References:

1. Real Time Systems – Jane W. S. Liu, Pearson Education Publication
2. Real Time Systems – Mall Rajib, Pearson Education
3. Real-Time Systems: Scheduling, Analysis, and Verification – Albert M. K. Cheng, Wiley.
4. RajkumarBuyya, SatishNarayanaSrirama, Fog and Edge Computing, Wiley Publications, 2019.
5. Wei Change and Jie Wu, Fog/Edge Computing for Security, Privacy and Applications, Springer, 2021.

COS-195: Parallel Computing Lab and Image Processing Lab **Lectures: 40**

M1: Parallel Computing Lab **[20L]**

Course Objectives:

To learn parallel programming with graphics processing units (GPUs). Outcomes: Students would learn concepts in parallel programming, implementation of programs on GPUs, debugging and profiling parallel programs.

CourseOutcomes:

Students would learn concepts of CUDA, implementation of programs,debugging and profiling programs in CUDA.

Detailed Syllabus:

Introduction : History, graphics processors, graphics processing units, GPGPUs. Clock speeds, CPU / GPU comparisons, heterogeneity. Accelerators, parallel programming, CUDA / OpenCL / OpenACC,

Hello World Computation: Kernels, launch parameters, thread hierarchy, warps / wavefronts, thread blocks / work groups, streaming multiprocessors, 1D / 2D / 3D thread mapping, device properties, simple programs

Memory: Memory hierarchy, DRAM / global, local / shared, private / local, textures, constant memory. Pointers, parameter passing, arrays and dynamic memory, multi-dimensional arrays.Memory allocation, memory copying across devices. Programs with matrices, performance evaluation with different memories

Synchronization: Memory consistency. Barriers (local versus global), atomics, memory fence. Prefix sum, reduction. Programs for concurrent data structures such as worklists, linked-lists. Synchronization across CPU and GPU
Functions: Device functions, host functions, kernels, functors. Using libraries (such as Thrust), developing libraries.
Support : Debugging GPU programs. Profiling, profile tools, performance aspects
Case studies : Image processing. Graph algorithms. Simulations.Deep learning.

M2: Image Processing Lab

Course Objectives:
To learn Image processing with MATLAB programming.
CourseOutcomes:
Students would learn concepts in image processing, implementation of programs on MATLAB, debugging and profiling programs.
Detailed Syllabus:

1. MATLAB program to extract different Attributes of an Image.
2. MATLAB program for Image Negation.
3. MATLAB program for Power Law Transformation.
4. MATLAB program for Histogram Mapping and Equalization.
5. MATLAB program for Image Smoothening and Sharpening.
6. MATLAB program for Edge Detection using Sobel, Prewitt and Roberts Operators.
7. MATLAB program for Morphological Operations on Binary Images.
8. MATLAB program for Pseudo Coloring.
9. MATLAB program for Chain Coding.
10. MATLAB program for DCT/IDCT Computation.

COS-196: Algorithm Lab using Python and Cloud Computing Lab Lectures: 40
M1: Algorithm Lab using Python [20L]
Course Objectives:

- To provide students with solid foundations to deal with a wide variety of computational problems.
- To provide a thorough knowledge of the most common algorithms so that these algorithms can be applied in different real life applications.

Course Outcomes:

- On completion of this course, the students will be able to:
- Analyze space complexity and time complexity of any algorithm.
- Implement any algorithm in python code.
- Apply these algorithms in some real life applications whenever required.

Details Syllabus:

1. Write a tail recursive version of a non tail recursive function.
2. Write a program to sort a list with Quiksort or Mergesort using divide and conquer strategy.
3. Implement a matrix chain multiplication problem using dynamic programming.
4. Write a program to find all pair shortest path of a given graph.
5. Write a program to solve 15 puzzle problem using branch and bound method.

6. Write a program to solve n-queen problem using backtracking.
7. Write a program to solve a given knapsack problem using greedy approach.
8. Write a program to implement BFS/DFS.
9. Write a program to find the shortest path of a given graph using Dijkstra's algorithm..
10. Write a program to implement RSA algorithm.
11. Write a program to implement Diffie-Hellman algorithm.

M2: Cloud Computing Lab [20L]

Course Objectives:

- To develop web applications in cloud
- To learn the design and development process involved in creating a cloud based application
- To learn to implement and use parallel programming using Hadoop

Course Outcome:

- On completion of this course, the students will be able to:
- Configure various virtualization tools such as Virtual Box, VMware workstation.
- Design and deploy a web application in a PaaS environment.
- Learn how to simulate a cloud environment to implement new schedulers.
- Install and use a generic cloud environment that can be used as a private cloud.
- Manipulate large data sets in a parallel environment.

Detailed Syllabus:

1) Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2) Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3) Install Google App Engine. Create hello world app and other simple web applications using python/java.
4) Use GAE launcher to launch the web applications.
5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6) Find a procedure to transfer the files from one virtual machine to another virtual machine.
7) Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8) Install Hadoop single node cluster and run simple applications like wordcount.

FIRST YEAR SECOND SEMESTER

M.Sc. (2nd Semester)

COS – 201: Advanced DBMS and Green Computing **Lectures: 40**
M1: Advanced DBMS **Lectures: 20**

Course Objectives:

- Understanding of the current theory and practice of database management systems.
- Providing a technical overview of database management systems.
- Get informed about data independence, integrity, security, recovery, performance, database design principles, and database administration.
- Getting knowledge about advanced topics like distributed and parallel database, object oriented database, deductive database and multimedia database.
- Understanding advanced transaction processing.

Course Outcomes:
At the completion of this course, students should be able to do the following:

- Understand the role of a database management system in an organization.
- Understand basic database concepts, including the structure and operation of the relational data model.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).
- Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
- Design and implement a small database project using SQL.
- Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.
- Describe and discuss selected advanced database topics, such as distributed and parallel database, object oriented database, deductive database and multimedia database.
- Get informed about advanced transaction processing.
- Understand WEB database and its applications.

Detailed Syllabus:
Introduction [**2L]**
Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS.
Relational Databases [**4L]**
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[2L]

Objects and Types, Specifying the behavior of objects, Implementing Relationships, Inheritance. Sample Systems. New Database Applications.

Multimedia Database [2L]

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 [2L]

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 [2L]

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 [3L]

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 [3L]

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.

Books:

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. 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.
7. Khoshafian: Object Oriented Databases, John Wiley & Sons,1993.
8. S. Khoshafian& A.B. Baker, Multimedia and Imaging Databases,Morgan Kaufmann Publishers, 1996.
9. Kemper &Moerkoette: Object-Oriented Database Management, PH, 1994.

M2: Green Computing **Lectures: 20**

Course Objectives:

Study the concepts related to Green IT, Green devices and hardware along with software methods, green enterprise activities, managing the green IT and various laws, standards, protocols along with the outlook of green IT.

Course Outcomes:

On successful completion of the course, the students will be able to attain below Course Outcome (CO):

- Discuss Green IT with its different dimensions and Strategies.
- Describe Green devices and hardware along with its green software methodologies.
- Discuss the various green enterprise activities, functions and their role with IT.
- Describe the concepts of how to manage the green IT with necessary components.
- Discuss the various laws, standards and protocols for regulating green IT.
- Identify the various key sustainability and green IT trends.

Detailed Syllabus:

Green IT: An Overview [3L]

Introduction, Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green IT, Holistic Approach to Greening IT, Greening IT, Applying IT for enhancing Environmental sustainability, Green IT Standards and Eco-Labelling of IT, Enterprise Green IT strategy, Green IT: Burden or Opportunity?

Green Devices and Hardware with Green Software [5L]

Green Devices and Hardware: Introduction, Life Cycle of a device or hardware, Reuse, Recycle and Dispose.

Green Software: Introduction, Energy-saving software techniques, Evaluating and Measuring software Impact to platform power.

Green Enterprises and the Role of IT[3L]

Introduction, Organization and Enterprise Greening, Information systems in Greening Enterprises, Greening Enterprise: IT Usage and Hardware, Inter-Organizational Enterprise activities and Green Issues, Enablers and making the case for IT and Green Enterprise.

Managing Green IT [3L]

Introduction, Strategizing Green Initiatives, Implementation of Green IT, Information Assurance, Communication and Social media.

Regulating the Green IT: Laws, Standards and Protocols [3L]

Introduction, The regulatory environment and IT manufacturers, Non regulatory government initiatives, Industry associations and standards bodies, Green building standards, Green data centers, Social movements and Greenpeace.

Green IT: An Outlook [3L]

Introduction, Awareness to implementations, Greening by IT, Green IT: A megatrend?, A seven-step approach to creating green IT strategy, Research and Development directions.

Books:

1. Harnessing Green IT Principles and Practices , San Murugesan, G.R. Gangadharan Wiley Publication, ISBN:9788126539680

COS- 202: Theory of Computer Science and Compiler Design **Lectures: 40**

M1: Theory of Computer Science

Course Objectives:

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.

Course Outcomes:

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

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

Detailed 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.

Books:

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.

M2: Compiler Design

Course Objectives:

The main Course Objectives 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.

Course Outcomes:

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

- To realize the 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 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.

Detailed Syllabus:

Introduction

About 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, Three address 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 .

Editor & Text Editor

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.

Books:

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. HenkAlblas 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: Artificial Intelligent and Soft Computing **Lectures: 40**

M1: Artificial Intelligent **Lectures: 20**

Course Objectives:

Students will try to learn:

- To create appreciation and understanding of both the achievements of AI and the theory underlying those achievements.
- To introduce the concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems
- To review the different stages of development of the AI field from human like behavior to Rational Agents.
- 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.

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

Course Outcomes:

Students will able to:

- Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
- Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
- Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing
- Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.
- Formulate and solve problems with uncertain information using Bayesian approaches.
- Apply concept Natural Language processing to problems leading to understanding of cognitive computing.

Detailed Syllabus:

Introduction [2L]

Course Introduction, Motivation.

Problem solving by search [4L]

State Space, Problem Reduction, Game Playing, Constraint Satisfaction.

Automated Reasoning [4L]

Proposition and first order logic, inference and deduction, resolution refutation, answer extraction, knowledge based systems, logic programming and constrained logic programming, non-monotonic reasoning.

Planning [3L]

State-space, plan space and partial order planning, planning algorithms.

Reasoning under uncertainty [3L]

Probabilistic reasoning, belief networks.

Learning [4L]

Inductive learning, decision trees, logical approaches, computational learning theory, neural networks, reinforcement learning, Intelligent agents, natural language understanding, Applications.

Books :

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, SecondEdition, Pearson Education
2. David Poole, Alan Mackworth, Randy Goebel, ”Computational Intelligence : a logical approach”, Oxford University Press.
3. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problemsolving”, Fourth Edition, Pearson Education.
4. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers.

M2: Soft Computing

Course Objectives:

The primary Course Objectives 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. It will provide the mathematical background for carrying out the optimization associated with neural network learning and 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.

Course Outcomes:

The student will be able to:

- Describe Artificial Neural Network (ANN)
- 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.

Detailed Syllabus:

Artificial Neural Networks [5L]

Basic-concepts-single layer perception-Multi layer perception-Supervised and unsupervised learning back propagation networks, Application.

Fuzzy Systems [5L]

Fuzzy sets and Fuzzy reasoning-Fuzzy matrices-Fuzzy functions-decomposition-Fuzzy automata and languages- Fuzzy control methods-Fuzzy decision making, Applications.

Neuro-Fuzzy Modeling [5L]

Adaptive networks based Fuzzy interfaces-Classification and Representation trees-Data dustemp algorithm –Rule base structure identification-nero-fuzzy controls

Genetic Algorithm [5L]

Survival of the fittest-pictures computations-cross over mutation-reproduction-rank method-rank space method, Application.

Books:

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.

COS- 204 (CBCS): Computer Fundamentals and Programming Concepts **Lectures: 40**

M1: Computer Fundamentals (CBCS) **Lectures: 20**

Course Objectives :

The Course Objectives of this course are

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

Course Outcome:

On completion of the course students will be able to

- Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming
- Write, compile and debug programs in C language and use different data types for writing the programs.
- Design programs connecting decision structures, loops and functions.
- Explain the difference between call by value and call by address.
- Understand the dynamic behavior of memory by the use of pointers.

Detailed 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.

M2: Programming Concepts (CBCS) **Lectures: 20**

Course Objectives:

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Be aware of the important topics and principles of software development.
- Have the ability to write a computer program to solve specific problems.
- Be able to use the Java SDK environment to create, debug and run simple Java programs.

Course Outcome:

After successful completion of the course, students can write and execute computer programs in C.

Detailed 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

Books:

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- 295: DBMS Lab and UML Lab **Lectures: 40**

M1: DBMS Lab

Course Objectives:

The key goal of ADBMS Lab is to incorporate, by query language programming, the theoretical principles of various topics of the database management system into reality. The students should increase their knowledge and develop their programming skills through the programming framework PL / SQL.

CourseOutcome:

- The students can implement databases, tables along with specifying data constraints.
- They will able to perform different operations on tables like insertion, deletion and update.
- Student can able to handle aggregate function along with JOIN operation.
- They will able to write view for enhancing database security.
- They can manipulate user permission of database access.
- They also will able to create procedure using PL/SQL.
- The student is also enabled to use of user interfaces and generating report using PL/SQL utilities.

Detailed Syllabus:

Experiments should include but not limited to :

1. Structured Query Language

- 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

3.Clause

- Using Aggregate Functions
- Combining Tables Using JOINS
- Subqueries
- Database Management
- Creating Views

- Creating Column Aliases
- Creating Database Users
- Using GRANT and REVOKE

4. Cursors in Oracle PL / SQL
5. Writing Oracle PL / SQL Stored Procedures.
6. Use of user interfaces and report generation utilities typically available with RDBMS products.
7. Developing mini projects based on PL/SQL

Books:

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

M2: Compiler Lab

Course Objectives:

- To implement Lexical Analyzer using Lex tool & Syntax Analyzer or parser using YACC Tool
- To implement NFA and DFA from a given regular expression
- To implement front end of the compiler by means of generating Intermediate codes.
- To implement code optimization techniques.

CourseOutcomes:

- To apply the knowledge of lex tool &yacc tool to develop a scanner & parser.
- To design & conduct experiments for NFA and DFA from a given regular expression
- To design & implement a front end of the compiler.
- To develop program for implementing symbol table.
- To develop program for solving parser problems.
- To create program for intermediate code generation.
- To learn the new code optimization techniques and apply it to improve the performance of a program in terms of speed & space.
- To learn & use the new tools and technologies used for designing a compiler
- To apply the knowledge of patterns, tokens & regular expressions in programming for solving a problem in the field of data mining.

Detailed Syllabus:

1. Write a program using LEX to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.
2.Write a program using LEX to recognize whether a given sentence is simple or compound.
3. Write a program using LEX to recognize and count the number of identifiers in a given input file.
4.Write a LEX program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.
5.Write a program using LEX to count the number of characters, words, spaces and lines in a given input file.
6. Program to find whether given number is Octal or Hexadecimal.

7.Program a C program to compute the FIRST of a given grammar.
8. Lex program to recognize keywords and identifiers.
9. Lex program to count number of vowels and consonant
10 Lex program to count the type of numbers
11. Lex program to count the number of printf and scanf statements
12. Lex program to find simple and compound statements
13. Lex program to count the number of identifiers
14.Lex program to count the number of words,characters,blank spaces and lines
15. Lex program to count the number of comment lines
16. Lex program to check the validity of arithmetic statement
17. Lex program to find the number of constants
18. Program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.
19. Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately
20. Program to recognize whether a given sentence is simple or compound.
21. Write a lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file

COS- 296: AI Lab and Soft Computing Lab

M1: AI Lab

Course Objectives:

- To study the applications of AI and agent based approach to AI.
- To study first-order predicate calculus, logical reasoning and problem solving using Prolog language.
- To study and discuss various techniques and algorithms of AI used in general problem solving, optimization problems, constraint satisfaction problems, and game programming.
- To familiarize students with various sub-areas of AI, such as expert systems, natural language processing and machine learning.

CourseOutcomes:

- Explain artificial intelligence, its characteristics and its application areas.
- Formulate real-world problems as state space problems, optimization problems or constraint satisfaction problems.
- Select and apply appropriate algorithms and AI techniques to solve complex problems.
- Design and develop an expert system by using appropriate tools and techniques.

Detailed Syllabus:

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.

M2: Soft Computing Lab

Course Objectives:

The course should enable the students to:

- Understand Fuzzy concepts
- Learn neural networks with back propagation and without preparation
- Learn the operators of genetic algorithms
- Practice on crisp partitions

CourseOutcomes:

- Explore methods that implements neural network techniques.
- Practice the fuzzy set relations using different operations.
- Design Regression techniques for a set of data points.
- Capture an appropriate classification model for analytical tasks.
- Implement best practices and techniques for computing efficiently.

Detailed Syllabus:

1. Write a program to implement Union, Intersection and Complement operations using Fuzzy logic.
2. Write a program to plot various membership functions.
3. Write a program to implement a Fuzzy Inference System.
4. Write a program to create fuzzy relation by Cartesian product of any two fuzzy sets and perform max-min composition on any two fuzzy relations.
5. Write a Program to implement De-Morgan's law using Fuzzy Logic.
6. Write a program to implement artificial neural network without back propagation.
7. Write a program to implement artificial neural network with back propagation.
8. Write a program for solving linearly separable problem using Perceptron Model.
9. Write a program to generate AND-NOT function using McCulloch-Pitts neural network.
10. Write a program to implement the classical Genetic algorithm.

SECOND YEAR FIRST SEMESTER

M.Sc. (3rd Semester)

COS-301: Advanced Networking and Network Security **Lectures: 40**

Course Objectives:

- Learning about computer network organization and implementation,
- Obtaining a theoretical understanding of data communication and computer networks,
- Gaining practical experience in installation, monitoring, and troubleshooting of current network systems.
- Introduces computer communication network design and its operations.
- Student should be able in part to design, implement, and maintain a typical computer network.

Course Outcomes:

- Provide security of the data over the network.
- Do research in the emerging areas of cryptography and network security.
- Implement various networking protocols.
- Protect any network from the threats in the world.
- Know the basic and advanced network operations and implementations.

Detailed Syllabus:

M1: Advanced Networking:

Layered communication architecture: layers, services, protocols, layer entities, service access

points, protocol function, Advanced Routing algorithms, Advanced Network Congestion Control algorithms, Quality of service, Real Time Transport Protocol, Internetworking, Performance Issues, Overview on VPN networks, Overview on Wireless Networks and Mobile Networks: LAN, PAN, Sensor Networks, Ad_hoc Networks, Mobile IP, Mobile TCP

IP Security.

M2: Network Security

Introduction

Attacks, Services and Mechanisms, Security Attacks, Security Services, Integrity check, digital Signature, authentication, has algorithms.

Secret Key Cryptography

Block Encryption, DES rounds, S-Boxes , IDEA: Overview, comparison with DES, Key expansion, IDEA rounds, Uses of Secret key Cryptography; ECB, CBC, OFB, CFB, Multiple encryptions DES.

Hash Functions and Message Digests

Length of hash, uses, algorithms (MD2, MD4, MD5, SHS) MD2: Algorithm (Padding, checksum, passes.) MD4 and 5: algorithm (padding, stages, digest computation.) SHS: Overview, padding, stages.

Public key Cryptography

Algorithms, examples, Modular arithmetic (addition, multiplication, inverse, and exponentiation)

RSA: generating keys, encryption and decryption. Other Algorithms: PKCS, Diffie-Hellman, El-Gamal signatures, DSS, Zero-knowledge signatures.

Authentication

Password Based, Address Based, Cryptographic Authentication. Passwords in distributed systems, on-line vs off- line guessing, storing. Cryptographic Authentication: passwords as keys, protocols, KDC's Certification Revocation, Inter- domain, groups, delegation. Authentication of People: Verification techniques, passwords, length of passwords, password distribution, smart cards, biometrics.

Security Policies and Security Handshake Pitfalls

What is security policy, high and low level policy, user issues? Protocol problems, assmptions,Shared secret protocols, public key protocols, mutual authentication, reflection attacks, use of timestamps, nonce and sequence numbers, session
keys, one-and two-way public key based authentication.

Network Security

Electronic mail security, IP security, Network management security. Security for electronic commerce: SSL, SET

System Security

Intruders and Viruses, Firewalls, Intrusion Detection

Books :

1- William Stallings, Wireless Communications & Networks, 2nd edition, Prentice-Hall Pearson, 2005

2- Jochen Schiller, Mobile Communication, (Latest edition), Addison Wesley

3- G. Wright and W. Stevens, TCP/IP Illustrated, Volume 2, Addison-Wesley, 1996.

4. William Stallings, "Cryptography and Network Security: Principals and

Practice", Prentice Hall, New Jersy.

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

6. Bruce Schiener, "Applied Cryptography".

COS-302: Machine Learning and Deep Learning **Lectures: 40**

M1: Machine Learning

Course Objectives:

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.

Course Outcomes:

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.

Detailed Syllabus:

Introduction

Machine learning, applications, concepts learning

Introduction to learning theory

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.

Books:

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.

M2: Deep Learning
Course Objectives:

It is a really challenging task to analyze and interpret huge volume of data. Deep Learning is one of the possible solutions to such tasks. The Course Objectives of the course is to make students familiar with not only basic concepts of Deep Learning but also modern Deep Learning concepts and architectures.

Course Outcomes:

On completion of the course, the students will know Deep Learning concepts and architectures and acquire the knowledge of applying these to solve various real life problems.

Detailed Syllabus:

Introduction

Fundamentals of Deep Learning, Multi-layer Perceptrons, Back propagation.

Convolutional Neural Networks (CNNs)

Basic concept of CNNs; CNN Architectures evolution: AlexNet, ZFNet, VGG, InceptionNets, ResNets, DenseNets; Visualization of Kernels.

CNNs in different applications

CNNs for Recognition and Verification; CNNs for Detection: Basic concept of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN,; CNNs for Segmentation: FCN, SegNet, U-Net, Mask-RCNN, etc.

Recurrent Neural Networks (RNNs)

Basic concept of RNNs; CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition.

Attention Models

Basic concept of Attention Models; Vision and Language: Image Captioning, Visual QA, Visual Dialog; Spatial Transformers; Transformer Networks; Vision Transformer

Deep Generative Models and its applications

Basic concept of Deep Generative Models: GANs, VAEs; Other Generative Models: PixelRNNs, NADE, Normalizing Flows, etc.

Recent Trends

Few-shot, One-shot, Zero-shot Learning; Self-supervised Learning; Reinforcement Learning; Other Recent Topics.

Books:

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, 2016
2. Michael Nielsen, Neural Networks and Deep Learning, 2016
3. YoshuaBengio, Learning Deep Architectures for AI, 2009

COS-303: Elective

A. Mobile Computing:
Course Objectives:

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.

Course Outcomes:

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.

Detailed 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.

Books:

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

B. Big Data Analytics **Lectures: 40**

Course Objectives:

Big data analytics helps any organization to harness the data and use it to identify new opportunities. It helps them to leads to smarter business moves, more efficient operations, higher profits. Businesses that use big data with advanced analytics gain value in many ways.

Course Outcomes:

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

1. understand how to harness the data and use it to identify new opportunities
2. significantly reduce costs when it comes to storing large amounts of data.
3. understand how to analyze the data to make faster and better decisions.

Detailed Syllabus:

Introduction to Big Data and Hadoop

Evolution of Big data - Big data characteristics - Big Data Use Cases - Characteristics of Big Data Applications - Perception and Quantification of Value -Understanding Big Data Storage - A General Overview of High-Performance Architecture - Analysing Data with Hadoop, Hadoop Streaming, HDFS - MapReduce and YARN - Map Reduce Programming Model

Hadoop Eco System

Pig : Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators.

Hive: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, Querying

Data and User Defined Functions.

Hbase :HBasics, Concepts, Clients, Example, Hbase Versus RDBMS.

Clustering and Classification

Overview of Clustering - K-means Clustering- - Overview of the Method - Determining the Number of Clusters - When to use - Cautions

Overview of Classification - Decision Tree - The General Algorithm - Decision Tree Algorithms - Evaluating a Decision Tree - Decision Trees in R - Naïve Bayes - Bayes Theorem - Naive Bayes Classifier.

Association and Recommendation System

Association Rules - Apriori Algorithm - Evaluation of Candidate Rules - Applications of Association Rules - Finding Association and finding similarity

Recommendation System: Collaborative Recommendation - Content Based Recommendation - Knowledge Based Recommendation- Hybrid Recommendation Approaches.

NoSQL Data Management

NoSQL Databases: Schema-less Models - Increasing Flexibility for Data Manipulation - Key Value Stores - Document Stores - Tabular Stores - Object Data Stores - Graph Databases

Review of Basic Data Analytic Methods using R.

Books:

1. Hadoop: The Definitive Guide by Tom White, O'reily Media
2. Intelligent Data Analysis by Michael Berthold, David J. Hand, Springer,
3. Big Data Analytics by SeemaAcharya, SubhasiniChellappan, Wiley.
4. Hadoop Operations by Sammer, O'reily Media
5. MapReduce Design Patterns by Miner & Shook, O'reily Media

C. Cryptography and Steganography **Lectures: 40**

Course Objectives:

- To understand basics of Cryptography and Steganography.
- To be able to secure a message over an insecure channel by various means.
- To learn about how to maintain data Confidentiality, Integrity and Availability

Course Outcomes:

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

- Provide security of the data over the network.
- Do research in the emerging areas of cryptography and steganography.
- Implement various networking protocols.
- Protect any network from the threats in the world.

Detailed Syllabus:

Introduction

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

Introduction, different Steam 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.

Books:

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

D. Bio Informatics **Lectures: 40**

Course Objectives:

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 livingsystems.
- Understand basic abstractions and computational approaches used to formulate and

address these problems.

Course Outcomes:

After undergoing the course, Students will be able to understand

- Sequencing Alignment and Dynamic Programming
- Sequence Databases

Detailed Syllabus:

Introduction

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.

Books:

1. M. Lesk, "Introduction to Bio Informatics," Oxford University Press
2. HoomanRashidi, 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

E. IoT and Embedded System **Lectures: 40**

Course Objectives:

This Course focuses on hands-on IoT concepts such as sensing, actuation and communication. It covers the development of Internet of Things (IoT) prototypes—including devices for sensing, actuation, processing, and communication—to help you develop skills and experiences. The Internet of Things (IOT) is the next wave, world is going to witness. Today we live in an era of connected devices the future is of connected things. To understand what is an Embedded System and then define it. Look at embedded systems from a historical point of view; Classify embedded systems; Look at certain applications & purposes of embedded systems

Course Outcomes:

- After the completion of the course, the students will be able design some IOT based prototypes,

- Understand IoT sensors and technological challenges faced by IoT devices, with a focus on wireless, energy, power, and sensing modules
- Market forecast for IoT devices with a focus on sensors
- Understanding of IoT value chain structure (device, data cloud), application areas and technologies involved.

- Explore and learn about Internet of Things with the help of preparing projects designed for Raspberry Pi
- Studying the embedded system they acquire a basic knowledge about fundamentals of microcontrollers
- Acquire a basic knowledge about programming and system control to perform a specific task.
- Acquire knowledge about devices and buses used in embedded networking
- Develop programming skills in embedded systems for various applications.
- Acquire knowledge about basic concepts of circuit emulators.

Detailed Syllabus:

Introduction to IOT [4L]

Understanding IoT fundamentals • IOT Architecture and protocols • Various Platforms for IoT • Real time Examples of IoT • Overview of IoT components and IoT Communication Technologies • Challenges in IOT, Domain Specific IoTs – Home, City, Environment, Energy, Agriculture and Industry.

Arduino Simulation Environment [5L]

Arduino Uno Architecture • Setup the IDE, Writing Arduino Software • Arduino Libraries • Basics of Embedded C programming for Arduino • Interfacing LED, push button and buzzer with Arduino • Interfacing Arduino with LCD

Sensor & Actuators with Arduino [5L]

Overview of Sensors working • Analog and Digital Sensors • Interfacing of Temperature, Humidity, Motion, Light and Gas Sensor with Arduino • Interfacing of Actuators with Arduino. • Interfacing of Relay Switch and Servo Motor with Arduino

Basic Networking with ESP8266 WiFi module [5L]

Basics of Wireless Networking • Introduction to ESP8266 Wi-Fi Module • Various Wi-Fi library • Web server- introduction, installation, configuration • Posting sensor(s) data to web server

IoT Protocols [3L]

M2M vs. IOT • Communication Protocols

Cloud Platforms for IOT [5L]

Virtualization concepts and Cloud Architecture • Cloud computing, benefits • Cloud services -- SaaS, PaaS, IaaS • Cloud providers & offerings • Study of IOT Cloud platforms • Thing Speak API and MQTT • Interfacing ESP8266 with Web services

Introduction to Embedded Systems [5L]

Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems. Core of Embedded Systems : Microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little endian processors, Application specific ICs, Programmable logic devices, COTS, sensors and actuators, communication interface, embedded firmware, other system components, PCB and passive components.

Characteristics and quality attributes of embedded systems [5L]

Characteristics, Operational and nonoperational quality attributes, application specific embedded system - washing machine, domain specific - automotive.

Programming Embedded Systems [3L]:

Structure of embedded program, infinite loop, compiling, linking and locating, downloading and debugging.

Books:

1. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
2. Internet of Things - A Hands-on Approach, ArshdeepBahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
3. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759
4. Programming Embedded Systems in C and C++, First Edition January, Michael Barr, O' Reilly Introduction to embedded systems, Shibu K V Tata McGraw-Hill.
5. Fundamentals of Microcontrollers and Applications In Embedded Systems" by ameshGaonkar
6. Embedded System Design by Peter Marwedel

F. Data Science **Lectures: 40**

Course Objectives:

- Apply quantitative modeling and data analysis techniques to the solution of real world business problems, communicate findings, and effectively present results using data visualization techniques.
- Recognize and analyze ethical issues in business related to intellectual property, data security, integrity, and privacy.
- Apply ethical practices in everyday business activities and make well-reasoned ethical business and data management decisions.

Course Outcomes:

- Demonstrate knowledge of statistical data analysis techniques utilized in business decision making.
- Apply principles of Data Science to the analysis of business problems.
- Use data mining software to solve real-world problems.
- Employ cutting edge tools and technologies to analyze Big Data.

- Apply algorithms to build machine intelligence.
- Demonstrate use of team work, leadership skills, decision making and organization theory.

Detailed Syllabus:

Introduction to Data Science [3L]

Definition of data science, importance, and basic applications, Understanding Exploratory Data Analysis

Statistics [5L]

Important statistical concepts used in data science; Difference between population and sample; Types of variables; Measures of central tendency; Measures of variability; Coefficient of variance; Skewness and Kurtosis

Inferential statistics [4L]

Normal distribution; Test hypotheses; Central limit theorem; Confidence interval ;T-test; Type I and II errors; Student's T distribution

Regression and Anova [3L]

Regression; ANOVA; R square; Correlation and causation

Exploratory data analysis [4L]

Data visualization ; Missing value analysis; The correction matrix; Outlier detection analysis; Data representation in form of a chart, diagram, plot, etc.

Machine Learning Algorithms [6L]

Using mathematical models or algorithms to recognize patterns, classification, or predictions Data Visualization

Data Analysis [3L]

Formatting or modeling data to discover insights using algorithms.

Coding (Python, R) [4L]

Basic coding to organize unstructured data.

Optimization Techniques [4L]

Optimization of software that is used in data extraction to gain the maximum output with limited resources.

Big Data [2L]

Managing enormous data sets to make extraction and data analysis easier.

Predictive Analysis [2L]

Use of data, algorithms, and models to predict outcomes based on historical data.

Books:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists Paperback by Andreas C. Mueller (Author), Sarah Guido (Author)
2. Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop (Author)
3. Python for Data: Data Wrangling with Pandas, Numpy, and Ipython Paperback by Wes Mckinney (Author)
4. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data Hardcover
5. R for data science: Import, Tidy, Transform, Visualize, And Model Data Paperback by Hadley Wickham (Author), Garrett Grolemund (Author)

G. Quantum Computing **Lectures: 40**

Course Objectives:

A basic introduction to quantum mechanics, linear algebra and familiarity with the Dirac notation is provided first to get one's quantum moorings right . This is then followed by an introductory treatment of quantum computation and quantum information covering aspects of quantum entanglement, quantum algorithms, quantum channels. Rudimentary quantum computing is introduced using the IBM quantum computer and associated simulators

Detailed Syllabus:

Introduction: Elementary quantum mechanics:, linear algebra for quantum mechanics,Quantum states in Hilbert space,The Bloch sphere, Density operators, generalized measurements, no-cloning theorem.
Quantum correlations: Bell inequalities and entanglement, Schmidt decomposition, superdense coding, teleportation.
Quantum cryptography: quantum key distribution
Quantum gates and algorithms: Universal set of gates, quantum circuits, Solovay-Kitaev theorem, Deutsch-Jozsa algorithm, factoring
Programming a quantum computer:The IBMQ, coding a quantum computer using a simulator to carry out basic quantum measurement and state analysis.

Text-books

1. Phillip Kaye, Raymond Laflammeet. al., An introduction to Quantum Computing, Oxford University press, 2007.
2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press,Cambridge, 2020
3. David McMahon-Quantum Computing Explained-Wiley-Interscience , IEEE Computer Society (2008)

H. Natural Language Processing **Lectures: 40**

Course Objectives:

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.

Course Outcomes:

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 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.

Text Books:

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

COS-304: (CBCS): DBMS and Internet Technology

M1: DBMS (CBCS) **Lectures: 20**

Course Objectives:

The Course Objectives 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 Outcomes :

- 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, relational database 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.

Detailed 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

M2: Internet Technology (CBCS) **Lectures: 20**

Detailed 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.

Books:

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-395: Advanced Networking Lab and Machine Learning Lab Lectures: 40

M-I: Advanced Networking Lab

Course Objectives:

- At the successful completion of this course, students will be able to:
- Implement a simple LAN with hubs, bridges and switches.
- Analyze the contents in a given Data Link layer packet, based on the layer concept.

Course Outcomes:

- 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.

Details Syllabus:

Experiments should include but not limited to :

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client-Server model
3. Write a code s simulating ARP/RARP protocols.
4. Write a code s simulating PING and TRACEROUTE commands
5. Create a socket for HTTP for web page upload and download.
6. Write a program to implement RPC (Remote Procedure Call)
7. Implementation of Sub-netting .
8. Applications using TCP Sockets like

a. Echo client and echo server

b. Chat

c. File Transfer

9. Applications using TCP and UDP Sockets like

a. DNS

b. SNMP
c. File Transfer
10. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS
11. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.
a. Link State routing
b. Flooding
c. Distance vector

Books:

1. Unix Network Programming: Networking APIs: Sockets and XTI, (Volume 1) by W. Richard Stevens, 2nd Edition, Prentice Hall India, 1999

M2: Machine Learning Lab

Course Objectives:

Introduce the concept of learning patterns from data and develop a strong theoretical and practical foundation for understanding state of the art Machine Learning algorithms. This course is broad in scope and gives the student a holistic understanding of the subject. This topic explores various paradigms for knowledge encoding in computer systems. Introduce subfields of AI such as Data Science, Recommender Systems, NLP, Game Playing, Bayesian Models, etc.

Course Outcomes:

On completion of the course students will be expected to:

- Have hands on knowledge on data collection, data generation, data repository building.
- Capacity on data cleaning and organize data for machine understandable format (.csv, .txt etc.)
- Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
- Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
- Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
- Be able to design and implement various machine learning algorithms in a range of real-world applications.

Instructions:

All experiments should be either in Python or Matlab or Weka platform.
The ML programs are not similar to traditional programs. So before going to implement ML programs you should know the installing libraries, packages, importing data, data cleaning, pre-processing, Training and Testing knowledge. The learner either would have to implement a minor project from any one of list of given assignments below or have to write a program on lottery basis in the laboratory.

List of assignments:

1. Implementing KNN Algorithm with Iris data set or any other .csv data.
2. Implementing decision tree Algorithm with some existing or new data set in the .csv format.
3. Implementing random forest Algorithm with some existing or new data set in .csv format.

4. Implementing random forest Algorithm with some existing or new data set in .csv format.
5. Implementing Naïve Bayes classifier Algorithm with some existing or new data set in .csv format.
6. Implementing Neural Network classifier Algorithm with some a data set in .csv format.
7. Implementing SVM classifier Algorithm with some data set in .csv format.
8. Implementing Linear Regression Algorithm with some data set in .csv format.
9. Implementing Logistic Regression Algorithm with some data set in .csv format.
10. Implementing spam and non-spam email filtering based on any classifier.
11. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
12. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
13. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
14. Implementing K-Means clustering with a data set taken from a repository.
15. Applying the PCA function into the training and testing set for analysis.
16. Apply SVD for compressing an image
17. Load a dataset and then find the TF-IDF of the said data
18. Implement CNN algorithm for image classification
19. Practical implementation of 1D CNN, 2D CNN
20. Sequential data representation using RNN/ LSTM

Books:

1. Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems Paperback – by DipanjanSarkar (Author), Raghav Bali (Author), Tushar Sharma (Author)
2. Machine Learning in Action by Peter Harrington ISBN 9781617290183
3. Practical Machine Learning: Tackle the real-world complexities of modern machine learning with innovative, cutting-edge techniques Paperback –by SunilaGollapudi (Author)

COS-396: M1: Industrial Tour, M2: Term Paper

The department will arrange an industrial visit for no less than three days and no more than one week, and students will be required to present a report on the trip, which will be assessed by a board of examiners appointed by the PG BOS.

SECOND YEAR SECOND SEMESTER

M.Sc. (4th Semester)

COS-491: Project Work

Each student will be assigned a different project that will be supervised by internal faculty members / external experts with permission to competent authority. The students will work with a supervisor assigned by the department committee to develop a project report that will be submitted to a board of examiners appointed by the PG BOS.

COS-492: Grand Viva

The students will appear for viva voce in front of the subject experts.

COS-493: Term Paper

Seminar topic will be assigned to individual students at the beginning of the semester.

Add-on / Value added course:

1. Agile Software Development (MySQL, PHP, Java)

Objective:

- Learn structured query language (SQL)
- Be able to write SQL queries to create, update, delete data.
- Be able to write SQL statements that create database objects.
- Understand the structure and design of relational databases.
- Understand the importance and major issues of database security and the maintenance of data integrity.
- Learn PHP programming to successfully build interactive, data-driven websites.
- Use the MVC pattern to organize code
- Test and debug a PHP application
- Work with form data
- Use cookies and sessions
- Work with regular expressions, handle exceptions, and validate data.
- Knowledge of object-oriented programming using Java
- Learn exception handling and multithreading using Java
- Learn Java Programming to build application and web apps.

Course Outcome:
On completion of this course, the students will be able to

- Summarize types of databases and how to design them.
- Describe database structures.
- Summarize advanced queries and advanced concepts in MySQL.
- Summarize managing users and privileges in MySQL.
- Write PHP scripts to handle HTML forms.
- Create PHP programs that use various PHP library functions, and that manipulate files and directories.
- Analyze and solve various database tasks using the PHP language.
- Analyze and solve common Web application tasks by writing PHP programs.
- Create programs in java using multithreading concept.
- Efficiently handle exception in any Java program.
- Write web apps using Servlet and JSP.

Detailed Syllabus:
MySQL:
Creating Tables
Creating tables, MySQL data types, Primary keys, Null values and Not Null Values, Auto increment
SQL Queries
Writing basic query, Use the Where clause, Conditional statements, Multiple conditions, Comparison Operators, Logic values, Null values, Wildcard characters, Compare column values, Advanced

queries, Distinct values, Top values, Aggregate functions, String functions, Sorting data, Ranking data

Grouping Data

Grouping data, Joins (Inner join, Left join, Full outer join, Self-join, Unions), Except and intersect

Advanced Uses

Importing and exporting data, Views, Locks and Transactions, MySQL Functions, Stored Procedures, Triggers, Generating Reports, Connection PHP and JSP with MySQL, performing basic database operation using a simple web page.

PHP:

Introduction

Basic Syntax, Defining variable and constant, PHP data type, operatorand expression; Decisions and loop; Function, call by value and call by reference, Recursive function; Strings, String searching, Replacing String, Formatting string, String related library functions; Index based and Associative array, using each () and foreach()

Working with files

File handling using PHP, Creating a File, Reading from Files, Copying Files,Moving File, Deleting File, Updating File, Uploading Files,

Webpage Design

Handling HTML form with PHP, Capturing form data, Generating file uploaded form, Redirecting a form after submission; Working with file and directories (creating, copying, renaming, deleting)

Session Management

Concept of Session and Cookie, using sessions and setting cookies with PHP, deleting cookies, registering session variables, Destroying the variables and Session.

Exception handling and Debugging

Understanding exception and error (Try, catch, throw), Error tracking and debugging.

Connection with Database

Introduction to RDBMS, Database connectivity with MySQL, Connection with MySQL, performing basic database operation (Insert, Delete, Update, Select), Setting query parameter, Executing query Join (Cross joins, Inner joins, Outer Joins, Self joins.),

Design a website that interacts with data in the database.

JAVA:

Introduction

Features of Java Language, JVM, Bytecode, Class Fundamentals, Object & Object reference, Garbage Collection, Constructor & initialization code block, Access Control, Modifiers, methods Nested, Inner Class &Anonymous Classes,Abstract Class & Interfaces Defining Methods, Method Overloading, Recursion, Dealing with Static Members, Finalize() Method, Native Method. Use of "this" reference, Use of Modifiers with Classes & Methods, Design of Accessors and Mutator Methods Cloning Objects, shallow and deep copying, Generic Class Types.

Array and string

Array and String, Defining an array, Initializing & accessing array, Multi-dimensional array, Operation on string, Mutable & immutable string, Collection classes, Loop for string, Tokenizing, StringBuffer, StringBuilder classes.

Inheritance

Benefits of Inheritance, Types of Inheritance in Java, Role of Constructors in inheritance, Overriding, Use of "super", Polymorphism in inheritance, Type Compatibility and Conversion implementing interfaces.

Package
Concept of package, Organizing classes and interfaces in packages, Package as access protection, Defining package, Packages import
Exception Handling
Idea behind Exception, Exceptions & Errors, Types of Exception, Control Flow in Exceptions, Use of try, catch, finally, throw, throws in Exception Handling, In-built and User DefinedExceptions, Checked and Un-Checked Exceptions.
Multithreading
Multithreading, Understanding threads, Needs of multi-threaded programming, Thread Priorities, Synchronization, Inter-thread communication, Deadlock.
Input/Output
Input/Output Operation in Java, Streams and the new I/O Capabilities, Understanding streams, The classes for input and output, The Standard streams, Working with file object, File I/O basics, Reading and writing to files, Buffer and buffer management, Read/Write operations with File channel, Serialization.
GUI Programming
Designing graphical user interfaces in Java, Components and containers, Layout Managers, AWT Components, Extending GUI features using Swing components, Java Utilities, The Collection, Event Handling.
Database Connectivity
Database Programming using JDBC, Introduction to JDBC, JDBC Drivers & Architecture, Connecting to non-conventional Databases.
Web-based Programming
Introduction to servlet and JSP, Servlet life cycle, Developing and Deploying Servlets, Handling Request and Response, Concept of JSP, Designing sample pages using servlet and JSP.

Textbooks / References:

1. Web Technologies, Black Book, Dreamtech Press
2. Learning PHP, MySQL, JavaScript, CSS and HTML 5, Robin Nixon, O'Reilly publication
3. MySQL: The Complete Reference, Vikram Vaswani, McGraw Hill Education
4. Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
5. Professional PHP Programming, Jesus Caspagnetto, Etal. Wrox Publication.
6. Internet and World Wide Web How to program, P.J. Deitel& H.M. Deitel, Pearson
7. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India
8. Java The Complete Reference, Herbert Schildt, Oracle Press
9. Programming with Java, E. Balagurusamy, McGrawHill
10. Java Servlets and JSP,Joel Murach and Michael Urban , Mike Murach & Associates

2. R Programming and Analytics

Objective:

- How to implement data structure in R
- R loop functions and debugging tools
- Writing custom R functions

- How to perform error handling
- The basics of statistical computing and data analysis
- How to use R for analytical programming
- Data visualization in R

Course Outcome:
On completion of this course, the students will be able to

- Explain critical R programming concepts
- Explain the use of data structure and loop functions
- Analyze data and generate reports based on the data
- Apply various concepts to write programs in R
- Generate graphs and plots depending on the given data
- Learn about statistical computing and data analysis

Detailed Syllabus:
Setting up R Environment
R Installation and Setting the R Environment to write programs in R
Exploring R Language
Variables, Operators & Data types; R data structures (Vector, List, Array, Matrix, DataFrame). Manipulation of the data structures (Subsettings, Merging, Column bind, Row bind, Melting, Casting etc.), Factors
Function in R
Functions & Control Flow Statements; String Manipulation Functions
Writing documents using Rmarkdown
Markdown, Knitr Package, writing documents using Rmarkdown using Knitr
Data Extraction
Data Extraction from CSV and Excel File, Data Extraction from URL, Clipboard, XML, JSON files, Data Extraction from Databases,
Data Exploration and Visualization
Understanding dplyr Package and using dplyr functions, Data Exploration, Data Visualization, Generating different types of charts and plots (Line Chart, Pie Charts, 3D Pie Charts, Bar Charts, Histogram, Scatter Plots etc.), Combining Plots
Machine Learning
Types of Machine Learning, Machine Learning Algorithms, K-Nearest Neighbor (KNN) Classification with suitable datasets, Naive Bayes Classification with suitable dataset, Decision Trees with suitable datasets, Support Vector Machine, Neural Networks & Random Forest, Regression (Linear Regression, Generalized Linear Regression, Non-Linear Regression, Logistic Regression), Clustering & K-Means Clustering

Textbooks / References:

11. R for Data Science - Garrett Grolemund and Hadley Wickham, O'Reilly
12. R Programming for Data Science - Roger D. Peng
13. Hands-on Programming with R - Garrett Grolemund, O'Reilly
14. Machine Learning with R - Brett Lantz, Packt Publishing