Syllabus of
Master of Science (M. Sc.) Computer Science
In effect from 2019-20

Raja Narendra Lal Khan Women's College (*Autonomous*),
Midnapore, West Bengal – 721 102
Affiliated to Vidyasagar University

FIRST YEAR FIRST SEMESTER

	Subjects	Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	* L	T	P	Theoretical / Practical Examination	Internal / Sessional	
CS-101	Data Structure using Python	3	1	0	40	10	4
CS-102	Computer Organization & Architecture	3	1	0	40	10	4
CS-103	Data Communication And Computer Network	3	0	0	40	10	3
CS-104	Switching & Finite Automata	3	0	0	40	10	3
CS-191	Data Structure using Python Lab	0	0	4	40	10	3
CS-192	Network Programming Lab	0	0	4	40	10	3
		12	2	8	240	60	20
	Total Period/Week = 22				**Total Marks = 300**		Total Credit=20

L – Lecture, T- Tutorial, P- Practical.

FIRST YEAR SECOND SEMESTER

	Subjects	Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
CS-201	Advanced Database Management System	3	0	0	40	10	3
CS-202	Design & Analysis of Algorithm	3	1	0	40	10	4
CS-203	Object-Oriented Programming with Java	3	0	0	40	10	3
CS-204 (CBCS*)	Programming Concepts with C Language	3	1	0	40	10	4
CS-291	DBMS Lab	0	0	4	40	10	3
CS-292	M-I: Object-Oriented Programming using Java Lab	0	0	2	20	5	3
	M-II: Computational Statistics in R Lab	0	0	2	20	5	
		12	2	8	240	60	20
	Total Period/Week = 22				**Total Marks = 300**		Total Credit=20

M-I: Module – I, M-II: Module-II.
CBCS – Choice Based Credit System, This paper is for opting by 2nd semester students from other PG departments.
Computer Science students will opt a CBCS paper offered by other PG departments.

SECOND YEAR FIRST SEMESTER

	Subjects	Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
CS-301	Advanced Operating System	3	1	0	40	10	4
CS-302	Principles of Compiler	3	0	0	40	10	3
CS-303	Elective-I	3	0	0	40	10	3
CS-304 * (CBCS)	Database Management System Concepts	3	1	0	40	10	4
CS-391	M-I: Operating System Lab	0	0	2	20	5	3
	M-II: Compiler Lab	0	0	2	20	5	
CS-392	Project-I	0	0	4	50	0	3
		12	2	8	250	50	20
	Total Period/Week = 22				**Total Marks = 300**		Total Credit: 20

Elective-I, Paper Code :CS-303

A. Graph Theory
B. Parallel Computing
C. Image Processing
D. Soft Computing
E. Cryptography

* This paper is for opting by 3rd semester students from other PG departments. Computer Science students will opt a CBCS paper offered by other PG departments.

SECOND YEAR SECOND SEMESTER

	Subjects	Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical / Practical Examination	Internal / Sessional	
CS-401	Machine Learning	3	1	0	40	10	4
CS-402	Elecctive - II	3	0	0	40	10	3
CS-403	Elecctive - III	3	0	0	40	10	3
CS-491	Machine Learning Lab using Python & MATLAB	0	0	4	40	10	3
CS-492	Project Work	0	0	8	0	80	5
CS-493	Grand Viva	0	0	0	0	20	2
		9	1	12	160	140	20
	Total Period/Week = 22				**Total Marks = 300**		

Elective – II, Paper Code: CS-402

A. Computational Geometry
B. Mobile Computing
C. Web Technology
D. Pattern Recognition
E. Computational Biology and Bioinformatics

Elective – III, Paper Code: CS-403

A. Data Mining
B. Cloud Computing
C. Data Science & Big Data
D. Computer Vision
E. Advanced Cryptography
F. Block Chain & Crypto Currency

M.Sc. (1st Semester)

CS-101: Data Structure using Python **Lectures: 40**

Module 1: Defining a Data Structures and types of Data Structures. Introduction to algorithms, and Big-O, Big-Ω , Big-θ Notations.
Module-2: Linear Structures: Array, Linked List, Stack, and Queue. Applications of arrays, lists, stacks and queues.

Module-3: Non-Linear Data Structures: Tree, Binary Tree, Tree Traversals, Applications of Trees, Binary Search Tree, Balanced Trees – AVL Trees, B-Tree. Graph – Graph representation, Graph traversal, Hashing and Collision Resolution Techniques.

Module-4: Searching and Sorting- Linear Search, Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Merge Sort, Quick Sort, and Heap Sort.
Trees – Binary Tree, Binary Search Tree, Tree Traversals.
Advanced Data Structures- Binomial Heaps, Fibonacci Heaps.

References:
1. Horowitz, Sahni, Fundamentals of Data Structures in C, Universal Press.
2. Mark Alan Weiss, Data Structures and Algorithm Analysis in C, Pearson Education.
3. Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms, PHI.
4. Horowitz, Sahni, Rajasekaran, Fundamentals of Computer Algorithms, Galgotia.

CS-102: Computer Organization & Architecture **Lectures: 40**

Module 1:
Introduction: Computer Architecture & Organization. Review basic combinational and sequential circuit. Von Neumann Architecture and Havard Architecture, Instruction Execution Cycle, Addressing Modes, Design of a simple ALU, bit-sliced ALU.

Module 2: Microprogrammed and Hardwired Control Unit, Memory organization: Primary memory, Secondary memory, Cache memory, Virtual memory.

Module-3: Basic Concepts of pipelining, Instruction Pipelining, Speedup. Hazards – data hazard, control hazard, structural hazard, Reservation Tables.

Module-4: CISC vs RISC, Basic Parallel Processing Architecture, Flynn's Taxonomy - SISD, MISD, SIMD, MIMD structures, von Neumann architecture vs Havard Architecture, Amdahl's Laws. Multiprocessor Architecture: Loosely Coupled & Tightly Coupled Systems, Vector Processor architecture, Introduction to Interconnection Network.

References:
1. Hennessy, Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann.
2. Hwang, Jotwani, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, TMH.
3. Flynn, *Computer Architecture: Pipelined and Parallel Processor Design*, Narosa Publishing.
4. Sivarama Dandamudi, *Guide to RISC Processors- for Programmers and Engineers*, Springer.
5. Sajjan G. Shiva, *Advanced Computer Architecture*, CRC Press.
6. Culler, Singh, Gupta, *Parallel Computer Architecture - A Hardware/Software Approach*, Morgan Kaufmann.

CS-103: Data Communication and Computer Network **Lectures: 40**

Module 1: Overview of data communication and Networking: Introduction; Data communications: components, Direction of data flow (simplex, half duplex, full duplex);
Analog modulation for analog signals: Amplitude modulation, Frequency modulation, Phase modulation.
Discrete analog modulation: Amplitude shift keying, Frequency shift keying, Phase shift keying, Differential Phase Shift Keying.
Digital modulation: Pulse-code modulation, Differential Pulse-code modulation.

Module 2: Networks Protocols and standards; Reference models: OSI reference model, TCP/IP reference model, their comparative study in brief.

Physical Layer: Data and signal fundamentals, Transmission impairments, Data rate limits for noisy and noiseless channels, Different line coding schemes, Block Analog to digital encoding, Analog Transmission, Concept of multiplexing, Frequency division multiplexing, Time division multiplexing

Data link layer: Types of errors, error detection & correction methods; HDLC; Multiple Access- Random Access, Controlled Access, Channelization. LANs and their Interconnection: Basic concepts, architectures, protocols, management and performance of Ethernet, token ring and token bus LANS; Repeaters and Bridges.

Module 3: Network Layer- Introduction, IPv4 Addressing, Routers, Routing Algorithms- Non Adaptive and Adaptive: OSPF, BGP; ICMP, Ipv6.

Module 4: Transport Layer- Introduction to Transport Layer Services, Connectionless Transport: UDP, Connection Oriented Transport: TCP, Flow control; Protocols: Stop and wait, ARQ, Go-Back- N, Selective repeat, Congestion Control, Sockets, Quality of services.

Application layer: DNS; SMTP, SNMP, FTP, HTTP & WWW, user authentication, Firewalls.

References:
1. Stallings, *Data and Computer Communication*, Prentice Hall of India.
2. Forouzan, *Data Communication and Networking*, McGraw-Hill.
3. Tanenbaum, *Computer Networks*, Prentice Hall.
4. Kurose, Ross, *Computer Network: Top-Down Approach*, Pearson Education.
5. Larry Peterson, Bruce Davie, *Computer Networks - a Systems Approach*, Morgan Kauffmann.
6. Taub, Schilling, Saha, *Principles of Communication System*, TMH.

CS-104: Switching Theory & Finite Automata
Lectures: 40

Module-1: Boolean algebra, switching functions, minimization of switching functions, Functionally complete switching functions. Sequential Circuit, Synchronous and Asynchronous sequential circuit.

Module-2: Alphabet, strings and languages. Definable languages, countable and uncountable sets, Cantor's theorem and proof that there are uncountably many languages, but only countable of them are definable.

Regular languages ad Finite Automata: deterministic and non-deterministic finite automaton (DFA and NFA). Equivalence of DFA and NFA. Minimization of DFA. Some closure properties of regular languages. Regular expressions, construction of NFA and DFA from a regular expression. Pumping lemma and existence of non-regular languages.

Module-3: **Context-free languages (CFL) and Pushdown Automata**: context-free grammar (CFG), push-down automaton (PDA), equivalence of CFG and PDA. Closure properties of CFLs. Chomsky normal form (CNF). Pumping lemma of CFL and exis-tence of non-context-free languages. Context-sensitive grammar and language.

Module-4: **Turing Machine**: Turing machine recognizable or recursively enumerable languages, recursive languages. Variants of Turing machine model and their equivalence. Decision problems of formal languages. Universal Turing machine. Language not recognized by Turing machine and undecidable problems of formal language. Mapping reducibility of decision problems.

Introduction to Complexity Classes: P, NP, NPC, NP-Hard.

References:

1. Michael Sipser, *Introduction to Theory of Computation*, Cengage Learning.
2. Hopcroft, Motwani & Ullman, *Introduction to Automata Theory, Languages, and Computation*, Pearson Education.
3. Z. Kohavi and N.K. Jha, *Switching and Finite Automata Theory*, TMH.
4. Mishra & Chandrasekaran, *Theory of Computer Science, Automata, languages and Computation*, PHI.
5. Lewis and Papadimitrou, *Elements of the theory of Computation*, PHI.
6. Linz Peter, *An Introduction to Formal Languages and Automata*, Narosa.

CS-191: Data Structure using Python Lab **Practicals: 40**

Experiments should include but not limited to :
Linked List manipulaion, Stack, Quesue, Recursive and Nonrecursive traversal of Trees,
Implementation of Binary Search Tree (BST) and node insertion and deletion,

AVL tree implementation.
Implementation of Linear Search, Binary Search, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap sort.
Implementation of Graph Traversals Algorithms- BFS, DFS.

Implementation of Single-source shortest path, All-pair shortest path, and Minimum Spanning Tree algorithms.

References:

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

CS-192: Network Programming Lab **Practicals: 40**

Experiments should include but not limited to :
Creation of sockets,
Exchange a character between TCP client & server,
Exchange of a string between TCP client & socket.
Exchange a character between UDP client & server,
Exchange of a string between UDP client & socket,
Implementation of broadcast of messages,
Implementation of multicast of messages.

References:

1. Donahoo, Calvert, *TCP/IP Sockets in C – Practical Guide for Programmers*, Morugan Kauffman.
2. Gay, *Linux-Socket Programming by example*, QUE Publication.

M.Sc. 2nd Semester

CS-201: Advanced Database Management System Lectures: 40

Module 1:

Introduction: Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS. E-R modelling.

Normalization: Pitfalls in RDBMS, importance of normalization, functional, multi-valued and join dependencies, INF to 5NF.

Module 2:

Database Tuning: Index selection and clustering, tuning of conceptual schema, denormalization, tuning queries and views.

Query Optimization: Importance of query processing, equivalence of queries, cost estimation for processing a query, general strategies, bi-relational and multi-relational join algorithms, algebraic manipulation.

Transaction and Concurrency Control in RDBMS: Transactions, Serializability, Testing for serializability, lock based and time-stamp based protocols; Deadlock detection and Recovery.

Module 3:

Object-oriented Databases: Objects and Types, Specifying the behavior of objects, Implementing Relationships, Inheritance.

Parallel and Distributed Databases: Distributed Data Storage: Fragmentation and Replication, Distributed Query Processing, Distributed Transaction Modeling and Concurrency Control, Distributed Deadlock, Commit Protocols, Design of Parallel Databases, Parallel Query Evaluation.

Introduction to Data Warehouse.

References:

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

CS-202: Design & Analysis of Algorithm Lectures: 40

Module 1: Time and space complexity of algorithms. Recursion and solution of recurrences. Algorithm Design Techniques: Greedy, Divide & Conquer, Dynamic Programming approaches. Greedy algorithms – greedy choice property, knapsack problem, 0-1 Knapsack problem, Minimum Spanning Tree algorithms and their time complexity.

Module 2: Divide and conquer algorithms - Merge sort, quick sort, and heap sort and their time complexity analysis, Selection of k^{th} largest element. Strassen's matrix multiplication algorithm.

Module 3: Graph Algorithms: Graph Traversals – BFS, DFS and their time complexity;
String Matching Algorithm - Knuth, Morris, Pratt (KMP) algorithm and its complexity.
Disjoint Set Maintenance Techniques.

Module 4: Dynamic programming algorithm – Matrix-chain multiplication, Single-source shortest path of graph, All-pair shortest path of graph and their complexity.

Introduction to Network Flow, Max-Flow Min-Cut theorem.
Introduction to P, NP-Complete and NP-Hard algorithms.

References:

1. Horowitz, Sahni, *Fundamentals of Data Structures in C*, Universal Press.
2. Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms*, PHI.
3. Horowitz, Sahni, Rajasekaran, *Fundamentals of Computer Algorithms*, Galgotia.
4. Kleinberg, Tardos, *Algorithm Design*, Pearson.

CS-203: Object-Oriented Programming using JAVA — Lectures: 20

Object Oriented System Development: Understanding Object Oriented Development, Understanding Object Oriented Concepts, Benefits of Object Oriented Development.

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

Applets and Event Handling.

GUI Design: Design GUI using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, text fields, menus.
Design Pattern: Implementation of Design Patterns.

Database Connectivity: Accessing and manipulating databases using JDBC.

References:

1. Rambaugh, James Michael, Blaha, *Object Oriented Modelling and Design*, Prentice Hall, India
2. Patrick Naughton, Herbert Schildt, *The Complete Reference - Java2*, TMH
3. Deitel and Deitel, *Java How to Program,* 6th Ed., Pearson
4. Ivor Horton, *Beginning Java 2 SDK* , Wrox.
5. Horstmann, Cornell, *Core Java Volume I – Fundamentals*, 9^{th} Edition, Prentice Hall.
6. Horstmann, *Core Java Volume II – Advanced Features*, Prentice Hall.
7. Erich Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson.

COS-204: Programming Concepts with C Language (CBCS) — Lectures: 40

Different types of programming languages, Built-in Data types, Constants & Variables, Input Output statements, Different type of operators, precedence and associativity of operators, Control Statements: if-else statement, switch-case cconstrut, Looping statements: while, do-while, for. Array, 2-D array. Function: declaration, definition and call. Call by value, and Call by address, Different header files and Library functions. Handling of character strings. Pointer. Dynamic memory allocation. Preprocessor Directives. Macro, Conditional Compilation. Command Line Arguments, Reading from and Writting to text file.

References:

1. B. W. Kernighan & D. M. Ritchie, *C Programming Language*, Pearson.
2. Yashavant P. Kanetkar, *Let Us C*, BPB Publications.
3. Balagurusamy, *Programming in ANSI C*, TMH.

COS-291: DBMS Lab — Practicals: 40

Module 1: Structured Query Language

I. *Creating Database*

Creating a Database, Creating a Table Specifying Relational Data Types, Specifying Constraints Creating Indexes

II. *Table and Record Handling*

INSERT statement, Using SELECT and INSERT together DELETE, UPDATE, TRUNCATE statements DROP, ALTER statements

III. *Retrieving Data from a Database*

The SELECT statement Using the WHERE clause,
Using Logical Operators in the WHERE clause,
Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING Clause,
Using Aggregate Functions, Combining Tables Using JOINS,
Subqueries

IV. *Database Management Creating*

Views Creating, Column Aliases, Creating Database Users Using GRANT and REVOKE Cursors

Module 2: PL / SQL

I. Writing Oracle PL / SQL Stored Procedures

II. Use of user interfaces and report generation utilities typically available with RDBMS products.

References

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, *Database System Concepts*, McGraw-Hill.
2. Raghu Ramakrishnan, *Database Management Systems*, McGraw-Hill.
3. Ivan Bayross, *SQL, PL/SQL the Programming Language of Oracle*, BPB Publication.

COS-292: M1: Object-Oriented Programming using Java Lab **Practicals: 20**

1. Assignments on class, constructor, overloading, inheritance, overriding
2. Assignments on wrapper class, vectors, arrays
3. Assignments on use of use of abstract class
4. Assignments on developing interfaces- multiple inheritance, extending interfaces
5. Assignments on creating and accessing packages
6. Assignments on multithreaded programming,
7. Assignments to manage errors and exceptions,
8. Assignments on applet programming and graphics programming
9. Assignments to develop simple GUIs.
10. Assignment to access database using JDBC

COS-292: M2: Computational Statistics in R Lab **Lectures: 20**

Working with integer, real, complex, and Boolean values.
Vectors and matrix operations.
System defined constants such as pi, Arithmetic and logical operators, elementary mathematical functions and statistical functions. Use of different packages.
Plotting graphs.
If-else statement, looping constructs.
Working with strings.
Inputs from user, and formatted output. Working with files.
User-defined functions.

Mean, Median, Variance, Standard deviation. ANOVA, random number generation, distribution of data set.
Introduction to data and visualization functions
Shaping data and using plyr, Exploring large data sets.

References:

1. Richard Cotton, *Learning R - A Step by Step Functional Guide*, O'Rielly.
2. Joseph Adler, *R in a Nutshell*, O'Rielly.
3. Norman Matloff, *The Art of R Programming- A Tour*, No Starch Press.

M.Sc. (3rd Semester)

CS-301: Advanced Operating System **Lectures: 40**

Module 1:
Hardware and OS overview: OS as a resource manager, virtual machine for every process; external and internal views; interaction with hardware architecture for protection and management of process resource, instruction set architecture (ISA);
Process: Its components and states, creation, and termination; parent and child process, loading process image, system calls and software interrupts; memory image of a process and kernel, logical and physical memory; kernel data structures related to process.
Thread: its components, creation and termination; relation with a process; user and kernel level threads and mapping; POSIX thread; race on global variables.
Context switching: system call, hardware interrupt and exceptions, their handling with hardware support.
Process Scheduling: preemptive and non preemptive scheduling, different measures of good performance; interrupt handler, scheduler and dispatcher; scheduling policies; multi-core processor scheduling;

Module 2:
Interprocess communication: message passing and shared memory; implementation as pipe, message queue, shared memory; problem of synchronization; signals.
Synchronization: concurrent and parallel execution, synchronization in cooperating processes, classical problems e.g. producer-consumer problem, reader-writer problem etc.; race condition and critical section of code (CS), protocol to access critical section, synchronization by busy wait (spin-lock), synchronization by process suspension; software solutions e.g. Peterson's algorithm, solution using machine level instructions e.g. test-and-set, exchange; semaphores, different usage, POSIX semaphore; synchronization and deadlock; condition variable and monitor.

Module 3:
Memory management: requirement, simple scheme with base-limit registers, problems of simple scheme; external fragmentation; swapping process image; segmented view of logical memory and mapping of of logical space using segment table; paging, mapping of logical space, page table, multi-level page table, paging hardware; inverted page table; context switching and memory mapping; demand paging, page fault, page replacement algorithms, dirty page and write-back; memory mapped files.

Module 4:
Disk Scheduling: SCAN, C-SCAN.
File and file system: file organization, directories, meta-data, links, storage management, allocation of storage space; special files e.g. device, file system as an interface; Linux file systems, inode, open-file and data structures, ext file systems; file allocation table (FAT); Page and Buffer Cache; journaling file system.
Deadlock detection and prevention.

References:

1. Silberschatz, Galvin, Gagne, *Operating System Concepts*, Wiley.
2. Stallings, *Operating Systems Internals and Design Principles*, PHI.
3. Daniel P Bovet & Marco Cesati, *Understanding the Linux Kernel*, O'Reilly.
4. Michael Kerrisk, *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, No starch press.

CS-302: Principles of Compiler **Lectures: 40**

Overview of the structure of a compiler. Different phases of compilation. Compilation for different targets, front-end and back-end. Idea of assembler, linker and library.

Lexical analyzer or scanner: Tokens and their attributes from the input character stream. Extended regular expressions to specify set of tokens.
Synthesis of finite state machine from the specification. *flex*: a lexical analyzer generator.

Parser or syntax analyzer: Specification of context-free structure of a language by CFG, context-sensitive features. Parse tree, syntax tree, different derivations, ambiguous grammar. Transformations of a grammar, removal of left-recursion and left-factoring. Idea of top-down and bottom-up parsing.

Top-down parsing: Computation of First() and Follow(). Recursive descent predictive parser and table driven predictive parser. Error recovery.

Bottom-up parsing: Dotted items, right-most derivation and automata of viable prefixes. Basic structure of shift-reduce parsing. Automaton of LR(0) items, LR(0) and SLR grammar and their parsing. Automaton of LR(1) items, LR(1) grammar, LALR grammar and their parsing.
Bison: A parser generator.
Use of ambiguous operator grammar. Error recovery

Semantic actions and synthesis of attributes: S and L attributed grammars. Syntax directed definition mars. Syntax directed definitions of different constructs of a programming language.

Intermediate representation: Annotated syntax tree, symbol table, 3-address code. Basic block and control-flow graph. SSA form. Intermediate code generation.

Code improvement and Target code generation: Peep hole, within basic block. Notion of data-flow analysis and its purpose. Register allocation. Target code generation.

References:
1. Aho, Lam, Sethi & Ullman, *Compilers: Principles, Techniques, & Tools*, Pearson Education.
2. Kieth D. Cooper & Linda Troczon, *Engineering a Compiler*, Morgan Kaufmann.
3. Grune, Reeuwijk, Bal, Jacobs & Langendoen, *Modern Compiler Design*, Springer.
4. John Levine, *Flex & Bison*, O'Reilly.

COS-303: Elective - I **Lectures: 40**

Graph Theory

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

Trees, Connectivity & Planarity: Spanning trees- Fundamental circuits, Spanning trees in a weighted graph, Cut sets – Properties of cut set, All cut sets, Fundamental circuits and cut sets, Connectivity and separability – Network flows, 1-Isomorphism, 2-Isomorphism, Combinational and geometric graphs, Planar graphs – Different representation of a planar graph.

Graph Coloring, Matching and Directed Graph: Graph Coloring: Four color problem , Chromatic Numbers and its bounds, Chromatic polynomial and its determination. Independence and Clique Numbers, Perfect Graphs- Definition and examples, Applications of Graph Coloring.

Matchings: Definitions and Examples of Perfect Matching, Maximal and Maximum Matching, Hall's Marriage Theorem.

Directed graphs: Types of directed graphs, Digraphs and binary relations, Directed paths and connectedness.

References:

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

Parallel Computing

Introduction: Parallelism in uniprocessor System, memory-interleaving, pipelining and vector processing, parallel computer structures, architectural classifications, parallel computer models: PRAM and VLSI complexity models, program properties: conditions of parallelism, program partitioning and scheduling, granularity and scalability.

Networked computers as a multi-computer platform, basics of message-passing, computing using work- station clusters, Software tools.

Parallel algorithms and their mapping on different architectures:
(i) Arithmetic computations: Addition, multiplication, FFT, DFT, Polynomial multiplication.
(ii) Matrix operations: Transpose, multiplication, inversion, eigenvalue computation.
(iii) Numerical applications: Solving systems of linear equations, finding roots of non-linear equations.
(iv) Sorting: Theoretical bounds, sorting networks, Batcher's odd-even and bitonic sort,
(v) Graph algorithms: All-pairs shortest-path (APSP) problem, finding connected components of a graph, minimum spanning tree.

References:

1. Grama, Gupta, Karypis, Kumar, *Introduction to Parallel Computing*, 2nd Edition, Addison-Wesley
2. Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann publication.
3. Victor Eijkhout, *Introduction to High PerformanceScientific Computing* Springer.
4. M.J. Quinn, *Parallel Programming in C with MPI and OpenMP*, McGraw-Hill

Image Processing

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 Transforms: 1-D and 2-D DFT, DCT, DST, Walsh, Hadamard, Haar, Slant, Karhunen-loeve - Singular value Decomposition transforms. Hough Transform, Generalized Hough Transform.
Image Restoration: Image degradation models, Weiner filter.
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, Eigen vector line fitting.

References:

1. Gonzalez, Woods, *Digital Image Processing*, Pearson.

2. Jahne, *Digital Image Processing*, Springer.
3. Chanda & Majumder, *Digital Image Processing & Analysis*, PHI
4. Jain, *Fundamentals of Digital Image Processing*, PHI
5. Sonka, *Image Processing, Analysis & Machine Vision*, VIKAS
6. Tinku Acharya, Ajoy Ray, *Image Processing - Principles and Applications*, Wiley.
7. Maria Petrou, Costas Petrou, *Image Processing - The Fundamentals*, Wiley.

SOFT COMPUTING

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

Fuzzy Set Theory - Definition, Different types of fuzzy set membership functions. Fuzzy set theoretic operations, Fuzzy rules and fuzzy reasoning, Fuzzy inference systems. Rough set theory.
Probabilistic Reasoning.
Genetic Algorithms, Simulated Annealing, applications.

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

Hybrid Systems and applications.

References:
1. Jang, Sun and Mizutani, *Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Printice Hall.
2. Tettamanzi, Andrea, Tomassini, and Marco, *Soft Computing : Integrating Evolutionary, Neural, and Fuzzy Systems*, Springer.
3. S. Y. Kung, *Digital Neural Network* , Prentice Hall International Inc.
4. Bart Kosko, *Neural Networks and Fuzzy Systems: Dynamical Systems Application to Machine Intelligence*, Prentice Hal

Cryptography

Introduction: Brief introduction to number theory, Euclidean algorithm, Euler’s totient function, Fermat’s theorem and Euler’s generalization, Chinese Remainder Theorem, primitive roots and discrete logarithms, Quadratic residues, Legendre and Jacobi symbols.

Introduction to Plaintext, Ciphertext, Encryption, Decryption, Key, Symmetric cipher, Asymmetric cipher, desirable properties of a crypto system.

Introduction to Substitution and Transposition cipher.
Stream Cipher: LFSR based stream cipher, RC4, Introduction to eSTREAM competition. HC-128 stream cipher.

Block Cipher: DES cipher, Triple-DES, IDEA cipher. Introduction to AES competition. AES cipher. Mode of Operation.

Public Key Cryptography: Diffie Hellman Protocol, RSA, El Gamal.

Introduction to Message Authentication.

Introduction to Message Integrity by Hash function. MD5 and SHA-1 hash functions. Introduction to SHA-2 competion.

Introduction to Digitial Signature.

References:

1. Stallings, *Cryptography and Network Security: Principals and Practice*,, Pearson Education.
2. Forouzan, Mukhopadhyaya, *Cryptography and Network Security*, TMH.
3. Stinson, *Cryptography Theory and Practice*, CRC Press.
4. Delfs, Knebel, *Introduction to Cryptography - Principles and Applications*, Springer.
5. "Understanding Cryptography" Paar, Pelzl, Preneel, Springer.

COS-304: DBMS (CBCS) Lectures: 40

Introduction: Concept & Overview of DBMS, Data Models, Database Languages, Database Administ rator, Database Users, Three Schema architecture of DBMS.

Entity-Relationship Model: Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets.

Relational Model: Structure of relational Databases, Relational Algebra, Relational Calculus,

SQL and Integrity Constraints: Concept of DOL, DML, DCL. Basic Structure, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Subqueries,

Database Normalization: Functional Dependency, Different anomalies in designing a Database. Normalization using functional dependencies, Decomposition, 1NF, 2NF, 3-NF and Boyce-Codd Normal Form.

Transaction processing: Concurrency control and Recovery Management: transaction model properties, state serializability, lock based protocols, two phase locking.

References:

1. Silberschatz, Korth, and Sudarshan, *Database System Concepts*, McGraw-Hill.
2. Ramakrishnan, *Database Management Systems*, McGraw-Hill.
3. Desai, *An Introduction to Database Systems*, Galgotia.

COS-391 M1: Operating System Lab Practicals: 20

Module 1:
Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).

Module 2:
Process: starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.
Semaphore: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).

POSIX Threads: programming with pthread functions(viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)

Inter-process communication: pipes(use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO)

References

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

COS-391 M2: Compiler Lab **Practicals: 20**

Familiarity with compiled codes in assembly language,
Writing a simple scanner,
Writing predictive parser for a small language,
Experiments with scanner (lex/flex) and parser (yacc/byson) generator (such as translation of regular expression to NFA or the construction or parse tree), Translation of the language to an intermediate form such as three-address code, Generation of target code in assembly language.

References:

1 Alfred Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman, "Compilers Principles, Techniques and Tools", Pearson Education Asia (2nd Ed. - 2009).
2Allen I. Holub "Compiler Design in C", Prentice Hall of India, 2003.
3. C. N. Fischer and R. J. LeBlanc, "Crafting a Compiler with C", Pearson Education.
4. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
5. John Levine, "Flex & Bison", O'Reilly.

COS-392 Project-I **Practicals: 20**

A separate project will be assigned to each student (or a group of two students) under the supervision of internal faculty members. This project work can be extended in Project-II in the final semester. The students will prepare a project report in consultation with the supervisor allotted by the department committee which will be presented before a board of examiners to be nominated by the B.O.S.

M.Sc. (4th Semester)

COS-401: Machine Learning **Lectures: 30**

Basics: Introduction to Machine Learning - Different Forms of Learning, Basics of Probability Theory, Linear Algebra and Optimization.

Regression Analysis: Linear Regression, Ridge Regression, Lasso, Bayesian Regression, Regression with Basis Functions.

Classification Methods: Instance-Based Classification, Linear Discriminant Analysis, Logistic Regression, Large Margin Classification, Kernel Methods, Support Vector Machines, Multi-class Classification, Classification and Regression Trees.

Neural Networks: Non-linear Hypotheses, Neurons and the Brain, Model Representation, Multi-layer Networks, Back-propagation, Multi-class Discrimination, Training Procedures, Localized Network Structure, Deep Learning.

Graphical Models: Hidden Markov Models, Bayesian Networks, Markov Random Fields, Conditional Random Fields.

Ensemble Methods: Boosting - Adaboost, Gradient Boosting, Bagging – Simple Methods, Random Forest.

Clustering: Partitional Clustering - K-Means, K-Medoids, Hierarchical Clustering - Agglomerative, Divisive, Distance Measures, Density Based Clustering – Dbscan, Spectral Clustering.

Dimensionality Reduction: Principal Component Analysis, Independent Component Analysis, Multidimensional Scaling, and Manifold Learning.

Reinforcement Learning: Q-Learning, Temporal Difference Learning

References:

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer.
2. R. O. Duda, P. E. Hart, and D.G. Stork, *Pattern Classification*, Wiley.
3. T. M. Mitchell, *Machine Learning*, McGraw-Hill.
4. Vladimir N. Vapnik, *Statistical Learning Theory*, Wiley.
5. Shawe-Taylor J., Cristianini N., *Introduction to Support Vector Machines*, University Press.
6. E. Alpaydin, *Introduction to Machine Learning*, PHI.

COS-402: Elective - II **Lectures: 40**

Computational Geometry

Geometric primitives, Convex Hulls (Graham's scan, Jarvis' March, Chan's algorithm), Plane Sweep and Incremental Construction paradigms, Point Location and trapezoidal maps, triangulation, line segment intersections, kd-trees, Binary Space Partition trees, Voronoi diagrams and Delaunay triangulations, Introduction to duality, Art Gallery theorem, Planar graphs, Euler's theorem and applications, Arrangements of lines and points, Zone theorem.

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

<u>Mobile Computing</u>

Introduction: Challenges in mobile computing, coping with uncertainties, resource scarcity, bandwidth, energy etc. Cellular architecture, co-channel interference, frequency reuse, capacity increase by cell splitting. Evolution of mobile system: CDMA, FDMA, TDMA, GSM.

Mobility Management: Handoff, types of handoffs; location management, HLR-VLR scheme, hierarchical scheme, predictive location management schemes. Mobile IP, cellular IP.

Publishing and Accessing Data in Air: Pull and push based data delivery models, data dissemination by broadcast, broadcast disks, directory service in air, energy efficient indexing scheme for push based data delivery.
Ad hoc and Sensor Networks: Routing Algorithms and Protocols, load-balancing, scheduling for reduced energy, coverage and connectivity problems.

References:
1. Raj Kamal, *Mobile Computing*, Oxford
2. Hansmann, Merk, Nicklous, Stober, *Principles of Mobile Computing*, Springer.
3. Jochen Schiller, *Mobile Communications*, Pearson Education
4. Stojmenovic and Cacute, *Handbook of Wireless Networks and Mobile Computing*, Wiley.

<u>Web Technology</u>

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

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

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

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

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

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

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

References:

1. Learning PHP, MySQL, JavaScript, CSS and HTML 5, Robin Nixon, O'Reilly publication
2. Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
3. Professional PHP Programming, Jesus Caspagnetto, Etal. Wrox Publication.
4. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson
5. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

Pattern Recognition

Introduction, overview of different approaches, decision boundaries, discriminant functions (linear and non-linear),
Bayesian classification, training and test sets,
Parametric and nonparametric learning, minimum distance classifiers, k-NN rule,
Unsupervised learning, basic hierarchical and non-hierarchical clustering algorithms, dimensionality reduction, similarity measures, feature selection criteria and algorithms, principal components analysis, some applications.

References:

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

Computational Biology and Bioinformatics

Introduction to molecular biology. Protein sequence data banks.
Sequence Alignments for DNA and protein sequences: Global alignments (Needleman-Wunsh), Local alignments (Smith-Waterman), k-mer based methods (BLAST), Advanced alignment methods (Gibbs sampling, suffix trees).
Genome: NOVA on genomics, Genetic mapping, Physical mapping, Recombinant DNA and Sequencing technologies, Whole-genome shotgun (Arachne) and clone-by-clone sequencing (Walking), Population genomics, SNP discovery, disease mapping, Gene recognition (Genscan) and cross-annotation (Rosetta).
Transcriptome and Evolution: Regulation— Transcription regulation, microarray technology, expression clustering,
DNA binding sites, location analysis, regulatory motif prediction, Ribozymes, RNA World, RNA secondary structure, non-coding RNAs, Evolution: RNA world, multiple alignments, phylogeny.
Protein Structure: Introduction to protein structure, Protein motifs.

References:

1. M. Lesk, *Introduction to Bio Informatics*, Oxford University Press
2. Hooman Rashidi, Lukas K. Buehler, *Bioinformatics Basics: Applications in Biological Science and Medicine*, CRC Press.

3. Jeffrey Augen, *Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and Information-Based Medicine*, Addison-Wesley.
4. Krawetz, Womble, *Introduction to Bioinformatics: A Theoretical and Practical Approach*, Humana Press.
5. Bryan Bergeron, *Bioinformatics Computing*, Prentice Hall.
6. Campbell, Heyer, *Discovering Genomics, Proteomics, and Bioinformatics*, Benjamin – Cummings.

COS-403: Elective – III Lectures: 40

Data Mining

Introduction: Introduction to data mining and knowledge discovery from databases. Scalability issues of data mining algorithms.
Introduction to Data warehousing: General principles, modelling, design, implementation, and optimization.
Data preparation: Preprocessing, sub-sampling, feature selection.
Classification and prediction: Bayes learning, discriminant analysis, decision trees, CART, C4.5 etc, neural learning, support vector machines, active learning. Combination of classifiers/ensemble learning.
Associations, dependence analysis, correlation, rule generation— a priori algorithm, FP Trees etc. and evaluation.
Cluster analysis and deviation detection: Partitioning algorithms, density based algorithms, hierarchical algorithms, model based algorithms, grid based algorithms, graph theoretic clustering etc.
Temporal and spatial data mining: Mining complex types of data. Visualization of data mining results.
Advanced topics: High performance computing for data mining, distributed data mining, soft computing tools for data mining.
Applications of data mining in bioinformatics, information retrieval, web mining, image and text mining.

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

Cloud Computing

Introduction: Introduction to Cloud Computing, Origin of Cloud Computing:
Fundamental concepts of Distributed Systems, Cluster Computing, Grid Computing, and Mobile Computing.

Cloud Models: Basics of Cloud Computing Concepts, Characteristics of Cloud Computing, Need for Cloud, Cloud Deployment models: private, public, hybrid and community cloud.

Cloud Services: Resource-as-a-Service (RaaS), Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS), Examples of each service. RaaS: Usage of Physical resources like servers, networks, data center etc,

IaaS: Virtualization, PaaS: Integrated lifecycle platform: Google App Engine, Microsoft Azure, Anchored life cycle platform: Salesforce platform,
SaaS: Characterizing SaaS, Salesforce's software environment.

Resource Scheduling for Cloud Computing: - Introduction, Virtual Machine provisioning and Migration Services, Scheduling techniques of Virtual machines for resource reservation, Cloud Service Scheduling hierarchy, Economic models for Resource-allocation scheduling , Heuristic Models for task –execution scheduling : Static Strategies , Dynamic Strategies , Heuristic Schedulers.

Cloud Security: Security and privacy issues of cloud.
Cloud Applications: Mobile Cloud Computing, Integration of Cloud with Wireless Sensor Network and its application.

References:

1. Kumar Saurabh, *Cloud Computing – Insight into New Era Infrastructure*, Wiley India.
2. John Rhoton, *Cloud Computing Explained*, Recursive Press.
3. Barry Sosinsky, *Cloud Computing Bible*, Wiley.
4. Rajkumar Buyya, James Broberg, *Cloud Computing: Principles and Paradigms*, Wiley.
5. Ju dith Hurwiz, *Cloud Computing for Dummies*, Wiley Publishing.

Data Science & Big Data

1. INTRODUCTION TO DATA SCIENCE AND BIG DATA
What is data science, relation to data mining, machine learning, big data and statistics.
Motivating examples
Why is it interesting?
Several data science settings
Introduction to the WEKA tool

2. GETTING TO KNOW YOUR DATA
From data to features: Representing problems with matrices, Representing problem with relations, Example - Text with TFIDF
Computing simple statistics: Means, variances, standard deviations, weighted averaging, modes, quartiles
Example- Political predictions
Simple visualizations: Histograms, Boxplots, Scatterplots, Time series, Spatial data
Case studies - X & Y examples, Medical data.

3. OVERVIEW OF TASKS & TECHNIQUES: PREDICTION
The prediction task: Definition, Examples, Format of input / output data
Prediction algorithms: Decision trees, Rule learners, Linear/logistic regression, Nearest neighbour learning, Support vector machines.
Properties of prediction algorithms and practical exercises.
Combining classifiers

4. EVALUATION AND METHODOLOGY OF DATA SCIENCE
Experimental setup: Training, tuning, test data; Holdout method, cross-validation, bootstrap method
Measuring performance of a model: Accuracy, ROC curves, precision-recall curves; Loss functions for regression
Interpretation of results: Confidence interval for accuracy; Hypothesis tests for comparing models, algorithms

5. DATA ENGINEERING
Attribute selection: Filter methods, Wrapper methods
Data discretization: Unsupervised discretization, Supervised discretization
Data transformations: PCA and variants.
6. OVERVIEW OF TASKS & TECHNIQUES: PROBABILISTIC MODELS
Introduction: Probabilities, Rule of Bayes and Conditional Independence
Naive Bayes: Application to spam filtering
Bayesian Networks: Graphical representation, Independence and correlation
Temporal models: Markov Chains, Hidden Markov Models

7. CASE STUDIES IN DATA SCIENCE & BIG DATA
Eve, the Pharmaceutical Robot Scientist: Data Science for Drug Discovery
Data science for sports analytics
Data science for sensor data (Introduction to challenge)

References:

1. Kotu, Deshpande, *Data Science: Concepts and Practice*, Morgan Kaufmann.
2. Sedkaoui, *Data Analytics and Big Data*, Wiley.
3. Igual, Seguí, *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*, Springer
4. Jesus Rogel-Salazar, *Data Science and Analytics with Python*, CRC Presss.
5. Simon Walkowiak, *Big Data Analytics with R*, Packt.
6. Kai Hwang, Min Chen, Big-Data Analytics for Cloud, IoT and Cognitive Computing, Wiley.

Advanced Cryptography

Elliptic Curve Cryptography.
Secret Sharing Schemes, Authentication codes.
Zero-Knowledge Proof, Homomorphic Encryption,
Identity based Encryption, Attribute-based Encryption.
Introduction to cryptanalysis: Cryptanalysis of Stream ciphers, Cryptanalysis of block ciphers – linear and differential cryptanalysis, Cryptanalysis of hash functions.

References:

1. B. Schneier, *Applied Cryptography*, Wiley.
2. Hankerson, Menezes, Vanstone, *Guide to Elliptic Curve Cryptography*, Springer.
3. Palash Sarkar Sanji Chatterjee, *Identity Based Encryption*, Springer.
4. Paar, Pelzl, Preneel, *Understanding Cryptography*, Springer.
5. Lars Knudsen, *The Block Cipher Companion*, Springer.
6. Avanzi, Doche, Tanja Lange, Nguyen, and Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press.
7. Nigel Smart, *Cryptography - an Introduction*, Springer.
8. Antoine Joux, Algorithmic Cryptanalysis, CRC Press.
9. Hinek, *Cryptanalysis of RSA and Its Variants*, CRC Press.
10. Paul, Maitra, *RC4 Stream Cipher and its Variants*, CRC Press.

Computer Vision

Introduction: Machine vision systems, optics and lenses, image sensors, human vision and Neurovisual model; Marr's paradigm; Imaging geometry— world co-ordinate system and camera coordinate system, co-ordinate transformations, projection geometry, camera calibration, radiometry.

Early processing and image filtering: Noise removal, region segmentation, concept of primal sketch, scale space, edge detection and localization, edge linking, Hough transform, corner and junction detection.
Reflectance map and photometric stereo: Image brightness and radiometry, image formation and surface reflectance under different conditions, reflectance map and bidirectional reflectance distribution function, photometric stereo recovering albedo and surface orientation, shape from shading.
Range measurement and recovering scene geometry: Binocular technique— stereo pair, epipolar line and plane, Stereo matching, photogrammetry, monocular technique— texture processing and shape from texture, depth from focusing and symmetry, different range finder (active) -laser range finder, light-stripe method.
Motion estimation: Motion field, optical flow— smoothness, boundary conditions, discontinuities of optical flow, block based method, pre-recursive method, Bayesian method, motion segmentation method, motion from points and lines, token tracking, stereo and motion tracking, use of Kalman filter, focus of expansion, structure from motion, motion compensated filtering and restoration, video compression, active and passive surveillance.
Representation and analysis of polyhedral scene: Understanding line drawings, gradient and dual space, generalized cylinder, volumetric representation, edge and junction labelling; Labelling and recognition of scene objects; Construction of model-base and visual learning, model based recognition system— Acronym, model based recognition from sparse range data, 3D model based vision system, scene understanding.

References:
1. Jahne, Haubecker, *Computer Vision and Applications for students and practitioners*, Academic Press.
2. Ritter, Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press.
3. Parker, *Algorithms for Image Processing and Computer Vision*, Wiley.

Block Chain & Crypto Currency

Blockchain: Introduction, Advantage over conventional distributed database, Blockchain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Blockchain application, Soft & Hard Fork, Private and Public blockchain.

Distributed Consensus: Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate.

Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin
.
Blockchain Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Blockchain.

References:
1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press.

COS-491 Machine Learning Lab using Python & MATLAB — Practicals: 20

The programs can be implemented in either Python or MATLAB.

Data sets can be taken from standard repositories (*https://archive.ics.uci.edu/ml/datasets.html*).

1. Implement and demonstratethe 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.
2. 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.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naive Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naive Bayesian Classifier model to perform this task. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regressionalgorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

References:

7. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer.
8. R. O. Duda, P. E. Hart, and D.G. Stork, *Pattern Classification*, Wiley.
9. T. M. Mitchell, *Machine Learning*, McGraw-Hill.

COS-492: Project-II

A separate project will be assigned to each student (or a group of two students) under the supervision of internal faculty members. This project work can be continuation of work done in Project-I. The students will prepare a project report in consultation with the supervisor allotted by the department committee which will be presented before a board of examiners to be nominated by the B.O.S.