Syllabus for 1st Sem CSE

1. Program Core I-MFCS

<table>
<tr><th>CODE</th><th>MFCS (4-0-0)
Syllabus</th></tr>
<tr><td colspan="2">Prerequisites:-
• Programming languages and data analysis tools
• Large-scale computation and the associated frameworks
• Mathematics and statistics and how machine learning builds on it.</td></tr>
<tr><td colspan="2">Course Objective: The enthusiastic Machine learning practitioner who is interested to learn more about the magic behind successful machine learning algorithms currently faces a daunting set of pre-requisite knowledge:
• Programming languages and data analysis tools
• Large-scale computation and the associated frameworks
• Mathematics and statistics and how machine learning builds on it.
Just like a solid foundation is essential to construct a building, MFCS forms an essential learning segment for machine learning. Linear algebra is an important field of mathematics and is essential for understanding how several machine learning algorithms work. Similarly, to understand randomness and probability of occurrence of one variable, one needs the knowledge of probability and stochastic process. Again, optimization algorithms help to understands the machine learning algorithms. Finally, MFCS has the following three pillars:
• Linear Algebra
• Probability & Stochastic Process
• Convex optimization</td></tr>
<tr><td colspan="2">Course Outcome:
After successful completion of this course, one will be able to understand:
• Relevance of linear algebra, probability, and Optimization in machine learning.
• Implementation of methods using the tools of linear algebra, optimization such as principal component analysis and linear least squares regression, support vector machines etc</td></tr>
<tr><td>Module-I</td><td>Linear Algebra: Finite dimensional vector spaces; Linear transformations and their matrix representations, rank, systems of linear equations, eigenvalues and eigenvectors, Cayley-Hamilton Theorem, diagonalization, Hermitian, Skew-Hermitian and unitary matrices; Finite dimensional inner product spaces,</td></tr>
<tr><td>Module-II</td><td>Linear Algebra: Row and Column spaces, Trace of a matrix, Special types of matrices, Decomposition of matrices- LU, LDU, UDU, QR, Gram-Schmidt, Cholesky, Singular value decomposition, Generalized inverse, Moore-Penrose inverse, Quadratic forms, Positive definite, Negative definite matrices, Matrix Calculus.</td></tr>
<tr><td>Module-III</td><td>Probability & Stochastic Process: Probability measure, Conditional probability, Random variables, Conditional expectation, Law of large numbers, Discrete & Continuous random variables, Stochastic Processes, The Poisson Process, Birth & Death Process, Markov chainsDiscrete & Continuous.</td></tr>
<tr><td>Module-IV</td><td>Convex optimization: Convexity, fundamentals of unconstrained optimization, line search method, gradient descent method, conjugate gradient decent method, Newton’s method, quasi-newton method, Levenberg-Marquardt method.</td></tr>
<tr><td>Suggested Books:</td><td>1. Probability, Statistics, and Queuing Theory with computer Science Applications, A.O. Allen, Academic Press (ELSEVIER), Indian Reprint
2. An Introduction to Probability Models, S. M. Ross, Academic Press (Elsevier-10th Edition), Indian reprint.
3. Introduction to Linear Algebra, G. Strang
4. Numerical Optimization, Nocedal and Wright.</td></tr>
<tr><td>Evaluation:</td><td>1. Assignments: 20%
2. Mid-term: 30%</td></tr>
</table>

	3. End-term Exam: 50%

2. Program Core II
Advanced DS and Algorithm

CODE	Advanced DS and Algorithm (3-1-0)
Prerequisites:- _UG level course in Data Structures and UG level course in Algorithm Design and Analysis	
Course Objective: This course introduces the basic concepts, principles, methods, implementation techniques, and applications of advanced data structure and algorithms. Course Objective: • CO1: The student should be able to choose appropriate data structures, understand the ADT/libraries, and use it to design algorithms for a specific problem. • CO2: To familiarize students with advanced paradigms and data structure used to solve algorithmic problems. • CO3: Introduce students to the advanced methods of designing and analysing algorithms and recent developments in algorithmic design. • CO4: The student should be able to understand different classes of problems concerning their computation difficulties, choose appropriate algorithms and use it for a specific problem.	
Course Outcome: Upon successful completion of this course, the student should learn: • Develop and analyse algorithms for symbol table, red-black trees, B-trees and Splay trees. • Develop algorithms for text processing applications and identify suitable data structures and develop algorithms for computational geometry problems. • Determine the appropriate data structure for solving a particular set of problems. • Categorize the different problems in various classes according to their complexity and develop an insight of recent activities in the field of the advanced data structure and algorithms.	
Module-I	Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries. Skip Lists: Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists. Trees: Red Black Trees, 2-3 Trees, B-Trees, Splay trees, Fibonacci Heaps, Van Emde Boas Priority Queues.
Module-II	ext Processing: Sting Operations, Brute-Force Pattern Matching, The BoyerMoore Algorithm, The Knuth-Morris-Pratt Algorithm, Rabin-Karp Fingerprinting Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm. Computational Geometry: One Dimensional Range Searching, Two Dimensional) 2 Range Searching, constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quadtrees, k-D Trees, Convex Hull, Line-segment Intersection. Sweep Lines.
Module-III	Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path. Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.
Module-IV	Linear Programming: Geometry of the feasibility region and Simplex algorithm, Formulation of Problems as Linear Programs. Duality. Simplex, Interior Point, Ellipsoid Algorithms. Number Theory and Algebra: Preliminaries, Polynomial roots and factors, Primality testing (Proof of PRIMALITY $\in$ NP) NP-completeness:

	Examples, proof of NP-hardness and NP-completeness. Approximation algorithms: Greedy Approximation Algorithms, Vertex Cover, Wiring, TSP
Suggested Books:	1. Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest.. Introduction to Algorithms, Prentice Hall of India Publications. 2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004. 3. Algorithm Design by Kleinberg and Tardos, Pearson. 4. The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman. 5. Merc De-Berg eta al. Computational Geometry: Algorithms and Applications, 3 rd Edition, Springer.
Evaluation:	1. Assignments: 20% 2. Mid-term: 30% 3. End-term Exam: 50%

3. Program Core III

CSXXX	ADVANCED COMUPTER ARCHITECTURE
Credits: 4 (4-0-0)	
Prerequisites for this course:	Basic knowledge of computer organization and architecture.
Course Objective:	This course focuses on advanced computer architectures such as pipelined and Multiprocessor systems, and low-level system software. **Course Objectives:** • An understanding of the fundamental computer architectural issues and the inherent limitations of the traditional approaches. • Familiarity with the principles and the terminologies involved in computer architecture, organization, and design. • Introduction to methods of specification, description, measurement and evaluation of processors and systems. • An appreciation of the historical developments in computer architecture and an acquaintance with many of the current innovative designs, providing a basis for understanding the new computer architectures that are on the horizon.
Course Outcome:	A student who has successfully completed this course should be able to • Understand the advanced concepts of modern computer architecture. • Analyze hardware & software trade-offs to design the instruction set architecture (ISA) interface. • Investigate modern design structures of Pipelined and Multiprocessors systems. • Analyze various performance characteristics of a computer system. • Apply knowledge of processor design to improve performance in algorithms and software systems.

Syllabus	
Module-I **[12] Hours**	**Processor Architecture:** Evolution of Microprocessors, Instruction set processor design, Principles of processor performance, RISC and CISC architectures. **Advanced processor design:** Pipelining fundamentals, Instruction level parallelism, Pipeline reservation table, Pipeline hazards, Data and Control Hazards, Minimizing pipeline stalls, Dynamic scheduling, Pipeline reservation tables, Tomasulo's approach, Branch Prediction and Speculation, Performance evaluation of pipelined architecture, Superscalar and Super pipelined architectures.
Module-II **[08] Hours**	**Hierarchical memory technology:** Memory hierarchy, Cache memory management, Multi-level caches, Data and Instruction caches, Block Replacement Techniques and Write Strategy, Cache optimization techniques.
Module-III **[12] Hours**	**Multiprocessor Architecture:** Basic multiprocessor architecture, Shared memory and message passing, Data-level parallelism, Thread-level parallelism, Simultaneous multi-threaded architectures, Instruction fetch policies in multi-threaded architectures, Multi-core architectures, VLIW processor architectures. SIMD Array Processors, SIMD Interconnection Networks, Interconnection structures for multiprocessors, Bus-based, Crossbar switch, Multistage switching, Hypercube. Multiprocessor Scheduling Strategies.
Module-IV **[08] Hours**	**Cache Coherence:** Multiprocessor cache coherence problem, Cache coherence protocols, MSI, MESI, MOSI, MOESI.
Suggested Books:	1. J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 4th Edition, Morgan Kaufmann 2. Dezso Sima, Terence Fountain, and Peter Kacsuk, Advanced Computer Architecture: A Design Space Approach, by Addison Wesley 3. V.Rajaraman and C.Sivaramamurthy, Parallel Computers Architecture and Programming, PHI , 2000 4. Hwang & Jotwani, Advance Computer Architecture, TMH
Evaluation:	1. Quizzes: 15% 2. Mid Term: 30% 3. End Term Exam: 50% 4. Teacher's Assessment: 5%

4. Program Elective I
(Any PE paper from the POOL of PE)

==

5. Program Elective II

(Any PE paper from the POOL of PE)

6. Mandatory Course

Research Methodology and IPR

(2-0-0)

Syllabus

Prerequisites:-

Course Objective:

Course Outcome:

At the end of this course, students will be able to
- Understand research problem formulation
- Analyze research related information
- Follow research ethics
- Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
- Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering.
- Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

Module-I	Unit 1: Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations Unit 2: Effective literature studies approaches, analysis Plagiarism, Research ethics,
Module-II	Unit 3: Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee
Module-III	Unit 4: Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.
Module-IV	Unit 5: Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. Unit 6: New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.
Suggested Books:	• Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'" • Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction" • Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners" • Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007. • Mayall, "Industrial Design", McGraw Hill, 1992. • Niebel, "Product Design", McGraw Hill, 1974. • Asimov, "Introduction to Design", Prentice Hall, 1962. • Robert P. Merges, Peter S. Menell, Mark A. Lemley, " Intellectual Property in New

	Technological Age”, 2016. • T. Ramappa, “Intellectual Property Rights Under WTO”, S. Chand, 2008
Evaluation:	4. Assignments: 20% 5. Mid-term: 30% 6. End-term Exam: 50%

7. Audit Course
(Any Audit Course from the POOL of Audit courses)

==

8. Laboratory I

Advanced DS (0-0-2)

Syllabus

Prerequisites:-

UG level course in Data Structures

Course Objective:

• This course concentrates on the practical part of the course of Advanced Data Structure using C/C++ Programming Language. This course allows students to understand practically the Logical and physical representation of data, algorithms, complexity and efficiency, data Structure operations.

• Data structures are fundamental building blocks of algorithms and programs

Course Objective:

• CO1: To be familiar with good programming design methods, particularly Top-Down design, and analysis of Time complexity various algorithms from this perspective.

• CO2: To understand the study of the Topics related to courses like “Principles of Programming Languages”, “programming methodologies”, and “design & analysis of algorithms” through knowledge of Data Structures and help the students in mastering their applications in real software projects.

• CO3: To develop algorithms for manipulating stacks, queues, linked lists, trees, and graphs, Hashing Techniques, searching and sorting.

• CO4: To develop recursive algorithms as they apply to trees and graphs

Course Outcome:

Upon successful completion of this course, the student should learn:

• Understand the importance of structure and abstract data type, and their basic usability in different applications through different programming languages.

• Analyze and differentiate different algorithms based on their time complexity.

• Understand various data structure such as stacks, queues, trees, graphs, etc. to solve various computing problems.

• Implement various kinds of searching and sorting techniques and know when to choose which technique to solve a real world problem.

Experiment No. 1	a. Write a program that use both recursive and non-recursive functions to perform the Binary search operation for a key value in a given list of integers. b. Write a program that implement Quick Sort, Merge Sort and Heap Sort method to sort a given list of integers in ascending order. c. Write a program that implement Quick Sort by taking median as the pivot element to sort a given list of integers in ascending order.
Experiment No. 2	Write a program that uses functions to perform the following operations on Binary Search Tree: i) Creation ii) Insertion iii) Deletion iv) Traversal (Inorder, Preorder, Postorder)
Experiment No. 3	Write a program that uses Stack operations to perform the following: a. Converting infix expression into postfix expression

	b. Evaluating the postfix expression.
Experiment No. 4	a. Write a program that uses functions to perform the insert and delete operations on an AVL Tree. b. Write a program that uses functions to perform the insert and delete operations on a B-Tree.
Experiment No. 5	a. Write a program to generate a graph given in Simple Interaction Format (SIF). b. Find single source shortest path (given a source) using Dijkstra algorithm. c. Find all pair shortest path using Floyd Warshall algorithm.
Experiment No. 6	Write a program to find minimum spanning tree using following algorithm from a graph given in SIF format. a) Prim's b) Kruskal
Experiment No. 7	Write a program to encode data in a text file using Huffman coding.
Experiment No. 8	Write a program to find longest common subsequence and all common subsequence between pair of string
Experiment No. 9	Write a program to implement Knuth-Morris-Pratt Algorithm and Rabin-Karp Fingerprinting Algorithm
Experiment No. 10	Write a program to multiply two matrices using Strassen's algorithm.
Suggested Books:	1. Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest.. Introduction to Algorithms, Prentice Hall of India Publications. 2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004. 3. Pai: "Data Structures & Algorithms; Concepts, Techniques & Algorithms "Tata McGraw Hill. 4. "Fundamentals of data structure in C" Horowitz, Sahani& Freed, Computer Science Press.
Evaluation:	1. Continuous Evaluation: 70% 2. End Term Assessment: 30%