Birla Institute of Technology, Mesra, Ranchi

BTECH IN ARTIFICIAL INTELLIGENCE

NEW COURSE STRUCTURE - To be effective from academic session 2022 - 23
BTECH IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
Based on CBCS system & OBE model Recommended scheme of study
(For Circuit Branches)

Course Level	Semester of Study	Course Code	Course Name	**Mode of delivery & credits** *L-Lecture; T-Tutorial; P-Practicals*			**Total Credits** *C-Credits*
				L (*Periods/week*)	**T *(Periods/week)***	**P *(Periods/week)***	**C**
FIRST	**FS**	MA103	Mathematic S-I	3	1	0	4
		CH101	Chemistry	3	1	0	4
	GE	EC101	Basic of Electronics and Communication Engineering	3	1	0	4
		ME101	Basic of Mechanical Engineering	3	1	0	4
	FS	CE101	Environmental Sciences	2	0	0	2
	LABORATORIES						
	FS	CH102	Chemistry Lab	0	0	3	1.5
	GE	EC102	Electronics and Communication Lab	0	0	3	1.5
		ME102	Engineering Graphics	0	0	4	2
	MC	MC101/102/103 /104	Choice of: NCC/NSS/ PT &Games/ Creative Arts(CA)	0	0	2	1
TOTAL(Theory+Labs)							**24**

SECOND	**THEORY**						
	FS	MA107	Mathematics-II	3	1	0	4
		PH113	Physics	3	1	0	4
		BE101	Biological Science for Engineers	2	0	0	2
	GE	CS101	Programming for problem Solving	3	1	0	4
		EE101	Basics of Electrical Engineering	3	1	0	4
	LABORATORIES						
	HSS	MT132	Communication Skills-I	0	0	3	1.5
	FS	PH114	Physics Lab	0	0	3	1.5
	GE	CS102	Programming for Problem Solving Lab	0	0	3	1.5
		PE101	Workshop Practice	0	0	3	1.5
	MC	MC105/106/107 /108	Choice of: NCC/ NSS/ PT & Games/ Creative Arts(CA)	0	0	2	1
TOTAL (Theory + Labs)							**25**
THIRD	**THEORY**						
	PC						
		MA205	Discrete Mathematics	3	1	0	4
		EC203	Digital System Design	3	0	0	3
		CS231	Data Structures	3	1	0	4

		CS233	Object Oriented Programming and Design Pattern	3	0	0	3
		CS235	Computer Organization and Architecture	3	0	0	3
	LABORATORIES						
	PC	EC204	Digital System Design Lab	0	0	3	1.5
		CS232	Data Structures Lab	0	0	3	1.5
		CS234	OOPDP Lab	0	0	3	1.5
	GE	EE102	Electrical Engineering Lab	0	0	3	1.5
	MC	MC201/202/203/204	Choice of: NCC/NSS/ PT & Games/CreativeArts(CA)	0	0	2	1
TOTAL(Theory+Labs)							**24**
THEORY							
FOURTH	**PC**	AI201	Probability and Statistics	3	0	0	3
	HSS	MT131	UHV2: Understanding Harmony	3	0	0	3
	PC	AI203	Mathematic S for Data Science	3	0	0	3
		CS241	Design and Analysis of Algorithms	3	0	0	3
		AI205	Introduction To Artificial	3	0	0	3

			Intelligence				
	OE		Open Elective-I	3	0	0	3
	LABORATORIES						
	PC	AI202	IT Systems Workshop (LEX,YACC)	0	0	2	1
		CS242	Design and Analysis of Algorithms Lab	0	0	2	1
		CS240	Shell and Kernel Lab	0	0	3	1.5
		AI204	Mathematics for Data Science Lab	0	0	3	1.5
	MC	MC205/206/207/208	Choice of: NCC/NSS/ PT& Games/Creative Arts(CA)	0	0	2	1
TOTAL(Theory+Labs)							**24**
FIFTH	**PC/PE**						
	PC	IT333	Data Comm. &Computer Networks	3	0	0	3
		CS361	Database System Concepts	3	0	0	3
		AI301	Supervised Learning	3	0	0	3
	PE		PROGRAME LECTIVE-I	3	0	0	3
			PROGRAME LECTIVE-II	3	0	0	3
	OE		Open Elective-II	3	0	0	3
	LABORATORIES						

	PC	IT334	Data Comm.& Computer Networks Lab	0	0	3	1.5
		CS238	Database Management System Lab	0	0	3	1.5
		AI302	Supervised Learning Lab	0	0	3	1.5
			PROGRAME LECTIVE LAB-II	0	0	3	1.5
TOTAL (Theory + Labs)							**24**
THEORY							
SIXTH	**PC/PE**						
	PC	AI303	Unsupervised Learning	3	0	0	3
		AI305	Deep Learning	3	0	0	3
		AI307	Modern Artificial Intelligence	3	0	0	3
	PE		PROGRAM ELECTIVE-III	3	0	0	3
	OE		Open Elective-III	3	0	0	3
	HSS	MT204	Constitution Of India	2	0	0	NC
	PROJ	MC300	Summer training				2
	LABORATORIES						
		AI304	Unsupervised Learning Lab	0	0	3	1.5
		AI306	Deep Learning Lab	0	0	3	1.5
	HSS	MT133	Communication Skills-II	0	0	3	1.5
TOTAL(Theory+Labs)							**21.5**
THEORY							
SEVE NTH							
	PC	AI401	Reinforcement Learning	2	0	0	2

	PE		PROGRAM ELECTIVE-IV	3	0	0	3
			PROGRAM ELECTIVE-V	3	0	0	3
	OE		Open Elective-IV	3	0	0	3
	PROJ	AI400M	Minor Project				3
	LABORATORIES						
	PE		PROGRA MELECTI VE LAB-IV	0	0	3	1.5
TOTAL (Theory + Labs)							**15.5**
EIGH TH	**PROJ**	AI400	Research Project /Industry Internship				**10**
GRANDTOTAL							**168**

***RequirementofProgrammeElectiveCourses(Theory/Lab):18creditorabove**

List of Program Electives(PE)

PE/LEVEL		**Code no.**	**Name of the PE Courses**	**Prerequisites/Corequisites**	**L**	**T**	**P**	**C**
		CS331	Formal Language and Automata Theory	MA205: Discrete Maths	3	0	0	3
3	**PE1**	CS362	Operating system concepts	CS231: Data Structures	3	0	0	3
3		IT337	Software Engineering	CS233: OOPDP	3	0	0	3
3		IT331	Image Processing	AI305, CS231	3	0	0	3
	PE2	AI309	Evolutionary Commutating	CS241: Design and Analysis of Algorithms	3	0	0	3
		AI310	Evolutionary Computing Lab	AI309: Evolutionary Computing	0	0	3	1.5
		AI311	Network Analysis	MA205, CS241	3	0	0	3
		AI312	Network Analysis Lab	AI311: N/w Analysis	0	0	3	1.5
		IT353	Block chain Technology	CS241	3	0	0	3
		IT354	Block chain Technology Lab	IT353	0	0	3	1.5
		AI313	Classical Optimization Techniques	CS241	3	0	0	3
		AI314	Optimization Techniques Lab	AI313	0	0	3	1.5
	PE3	AI315	Advanced Algorithms	CS241	3	0	0	3
		AI317	Information Retrieval	CS241	3	0	0	3
4		AI319	Introduction to Compiler Design	CS331	3	0	0	3
		AI321	Data Mining	CS237: DBMS	3	0	0	3
4	**PE4**	IT347	Introduction to Distributed Systems	CS235, IT333	3	0	0	3
		IT348	Distributed Computing using SPARK Lab	IT347	0	0	3	1.5
4		IT445	Internet of Things (IoT)	IT333: DCCN	3	0	0	3
		IT446	Internet of Things Lab	IT445	0	0	3	1.5

4		CS437	Deep Learning	3	0	0	3	4
		AI425	Computer Vision	IT331, AI305	3	0	0	3
		AI426	Computer Vision Lab	AI306: Deep Learning Lab, AI425	0	0	3	1.5
		IT451	Cloud Computing	IT333: DCCN	3	0	0	3
4		IT452	Cloud Computing Lab	IT451	0	0	3	1.5
	PE5(NO Lab)	AI427	Robotics	AI205, AI307: Modern AI	3	0	0	3
		IT351	Natural Language Processing	CS241	3	0	0	3
		AI429	Speech Processing	CS331: FLAT	3	0	0	3
		IT349	Cryptography & Network Security	CS241	3	0	0	3

\

SYLLABUS

SEMESTER III COURSE INFORMATION SHEET

Course code: MA205
Course title: Discrete Mathematics
Course code: MA 205
Course Name: Discrete Mathematics
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:1 P:0 C: 4
Class schedule per week: 3 Lectures, 1 tutorial
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	to model and analyze computational processes using analytic and combinatorial methods
2.	solve the problems of graph theory using graph algorithms
3.	apply computer programs (e.g. SAGE) to study graphs.
4.	apply counting techniques to solve combinatorial problems and identify, formulate, and solve computational problems in various fields.
5.	apply graph theory in the areas of computer science, operation research, biology, chemistry, physics, sociology, and engineering

Module 1: **[8 Lectures]**
Mathematical logic and Mathematical Reasoning, Compound Statements, Propositional Equivalences, Predicates and Quantifiers, Methods of Proof, Mathematical Induction, Well-ordering principal, Recursive Definition and Algorithms.

Module 2: **[8 Lectures]**
Recurrence Relations, Classification of Recurrence Relations and their solutions by Characteristic Root method, Generating function and their various aspects, Utility of Generating function in solving Recurrence Relations.

Module 3: **[8 Lectures]**
Set, Operations on Set, Computer representation of Set, Relations, Properties/Classification of Relations, Closure operations on Relations, Matrix representation of Relations, Digraphs. Functions and their Representation, Classification of Functions, Warshall's algorithm, Discrete Numeric Functions, Growth of Functions, Big O, Big Q, Hash Function, Growth Functions.

Module 4: **[8 Lectures]**
Binary Operations, Groups, Product and Quotients of Groups, Semi group, Products and Quotients of Semi groups, Permutation Group, Composition of Permutation, Inverse Permutation, Cyclic Permutation, Transposition, Even and Odd Permutation, Coding of Binary Information and Error Correction, Decoding and Error Correction.

Module 5: **[8 Lectures]**
Introduction to Graph, Graph Terminologies and their Representation, Connected & Disconnected graphs, Isomorphic Graph, Euler & Hamilton graphs. Introduction to Trees, Versatility of Trees, Tree traversal. Spanning Trees, Minimum Spanning Tree.

Text Books:

1. **Mott, Joe L., Abraham Kandel, and Theodore P. Baker** Discrete Mathematics for Computer Scientists & Mathematicians, PHI, 2^{nd} edition 2002.
2. **Swapan Kumar Chakraborty and Bikash Kanti Sarkar**: Discrete Mathematics, Oxford Univ. Publication, 2010.
3. **Kolman, Bernard, Robert C. Busby, and Sharon Ross.** Discrete Mathematical Structures, Prentice-Hall, Inc., 2003**.**

Reference Books:

1. Bikash Kanti Sarkar and Swapan Kumar Chakraborty, Combinatorics and Graph Theory, PHI,2016.

Course code: EC203
Course title: Digital System Design
Pre-requisite(s): EC101 Basics of Electronics Communication Engineering
Co- requisite(s):
Credits: L: 3 T:0 P:0 C:3
Class schedule per week: 3
Class: B. Tech
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	Explain the concept of digital electronics.
2.	Apply the knowledge to produce digital electronics circuits.
3.	Analyse and categorize digital circuits.
4.	Justify the uses of different digital circuits.
5.	Schematize and demonstrate simple computing machines.

Module 1: Basics of Digital Electronics [8 Lectures]
Number representation, Binary number system, Number base conversion, Octal, Hexadecimal and BCD codes, binary Arithmetic, Logic gates, Introduction to VHDL and Verilog, VHDL Models, Logic Families: TTL, ECL, and CMOS Logic Circuits, Logic levels, voltages and currents, fan-in, fan-out, speed, power dissipation. Comparison of logic families.

Module 2: Simplification of Boolean functions [8 Lectures]
Boolean Algebra, Basic theorems and Properties, De Morgan's theorem, Canonical & Standard forms, Simplification of Boolean function using Karnaugh map, POS& SOP simplification, Prime implicant, NAND and NOR implementation,

Module 3: Design of Combinational Circuits [8 Lectures]
Analysis and design procedure, Parity Generators and Checkers, Adders, Subtractors, Look ahead carry, Adder, 4-bit BCD adder/subtractor, Magnitude comparator, Decoders, Encoders, Multiplexers, De-multiplexers, , Design of 1 bit ALU for basic logic and arithmetic operations.

Module 4: Design of Sequential Circuits and Memories [8 Lectures]
Basic Latch, Flip-Flops (SR, D, JK, T and Master-Slave), Triggering of Flip Flops, Synchronous and asynchronous counters, Registers, Shift Registers, Memories and Programmable Logic design, Types of memories, Memory Expansion and its decoding, Programmable Logic Arrays (PLA), Programmable Array Logic (PAL)

Module 5: Design of simple computing machines [8 Lectures]
SAP-I concepts with stress on timing diagrams, Microinstructions, Fetch and Execution cycle variable machine cycle, Hardware control Matrix, Macroinstructions, Microprogramming , Bus concepts, Multiplexed Minimum system. Pipelining concepts.

Text Books:

1. "Digital Design", Morris Mano and Michael D. Ciletti ,5th edition PHI 2.
2. "Digital System Design using VHDL", Charles H Roth, Thomson Learning

Reference Books:

1.Digital computer Electronics AP Malvino, 3rd Edition Mc Graw Hill

Course code: **CS231**
Course title: **Data Structures**
Pre-requisite(s): **Programming for Problem Solving**
Co- requisite(s): **Data Structures Lab**
Credits: L: 3 T: 1 P: 0
Class schedule per week: **4**
Class: **B. Tech**
Branch**: AI/ML**

Course Outcomes

After the completion of this course, students will be:

1.	Define various linear and non-linear data structures like stack, queue, linked list, tree and graph.
2.	Explain operations like insertion, deletion, traversal, searching, sorting etc. on various data structures.
3.	Design various data structures and their operations.
4.	Analyze the performance of data structure based operations including searching and sorting.
5.	Justify the choice of appropriate data structure as applied to specified problem definition.

Module 1:Basic Concepts & Definitions **[8 Lectures]**
Data Structure, ADT, Algorithms, Time and Space Complexity, Asymptotic Notations (O, θ, Ω), Time complexity computation of non-recursive algorithms (like Matrix addition, Selection sort – using step count), Array – basic operations, concept of multi-dimensional array, Polynomial operations using Array, Sparse Matrix.

Module 2: Stacks & Queues **[8 Lectures]**
Stack ADT: basic operations, Queue ADT: basic operations, Circular Queue, Evaluation of Expressions, Another application or Mazing Problem.

Module 3: Linked Lists **[8 Lectures]**
Singly Linked List: concept, representation and operations, Circular Linked List, Polynomial and Sparse Matrix operations using LL, Doubly Linked List: basic concept.

Module 4: Trees & Graphs **[8 Lectures]**
Basic concepts and terminologies, Binary Search Tree and Heap, Disjoint Set, Graph: concept and terminologies, Concept of BFS, DFS, Spanning Tree, Connected Components.

Module 5: Searching & Sorting **[8 Lectures]**
Sequential Search and Binary Search, Insertion Sort, Heap Sort, Radix Sort, External Sorting: k-way merging approach.

Text Books:

1. Sahni Horwitz,, Freed Anderson, Data Structures in C, 2nd Edition (or latest) , University Press.

Reference Books:

1. Thareja Reema, Data Structures Using C, 2nd Edition, Oxford University Press.
2. Tanenbaum, Langsam, Augenstein, Data Structures using C, Pearson.

Course code: **CS233**
Course title: **Object Oriented Programming and Design Patterns**
Pre-requisite(s): **Data Structures**
Co- requisite(s):
Credits**: L: 3 T: 0 P: 0**
Class schedule per week: **3**
Class: **B. Tech**
Branch**: AI/ML**

Course Outcomes

After the completion of this course, students will be:

1.	Identify the difference between procedural and OO programming.
2.	Construct programs using various OOP principles.
3.	Design UI using JAVA GUI components.
4.	Operate on files and strings in real life scenarios.
5.	Analyze thread performance and inter thread communication issues

Module 1:Introduction to Classes, Objects and Java [8 Lectures]
Introduction to Object Technology, Java, Understanding the Java development environment, Programming in Java, Memory concepts, Doing basic Arithmetic, Comparing entities, Classes, Objects, Methods, Strings, Primitive vs reference types.

Module 2:Control Statements, Methods and Arrays [8 Lectures]
Basic selection statements, Iterative constructs, Relative and Logical operators, break, continue, Methods, static methods, parameter passing, argument promotion and casting, scopes, method overloading. Arrays and ArrayList in Java, Enhanced for statement, Passing arrays to methods, Multidimensional arrays, Using command line arguments.

Module 3: Object Oriented Concepts: Polymorphism & Inheritance [8 Lectures]
Controlling access to class members, the use of this keyword, getters and setters, Composition, enum, the use of static and final, Garbage collection. Superclass and subclass, protected members, constructors in subclass, the Object class, Introduction to polymorphism, Abstract classes and methods, Assignment between subclass and superclass variables, Creating and using interfaces.

Module 4: Exception Handling & GUI Design [8 Lectures]
When to use exception handling, Java exception hierarchy, finally block, Stack unwinding, Chained exceptions, Declaring new exception types, Assertions, try with resources. Simple I/O with GUI, Basic GUI Components, GUI Event handling, Adapter classes, Layout managers, Using panels.

Module 5: String, Characters & Files **[8 Lectures]**

Working with the String and StringBuilder class, Character class, Tokenizing strings, Regular Expressions, Files and Streams, Using NIO classes, Sequential file handling, Object serialization, JFileChooser, Introduction to threading, Introduction to Generics and lambda expressions.

Text Books:

1. Deitel P., Deitel H., Java How to Program, 10^{th} Edition, Pearson Publications, 2016

Reference Books:

1. Wu C. T., Object Oriented Programming in Java, 5^{th} Edition, McGraw Hill Publications, 2010

Course code: **CS235**
Course title: **Computer Organization Architecture**
Pre-requisite(s): **Digital Logic**
Co- requisite(s):
Credits: **L: 3 T: 1 P: 0**
Class schedule per week: 4
Class: B. Tech
Branch: CSE/IT/AIML

Course Outcomes

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

1.	Explain the merits and pitfalls in computer performance measurements and analyze the impact of instruction set architecture on cost-performance of computer design
2.	Explain Digital Logic Circuits ,Data Representation, Register and Processor level Design and Instruction Set architecture
3.	Solve problems related to computer arithmetic and Determine which hardware blocks and control lines are used for specific instructions
4.	Design a pipeline for consistent execution of instructions with minimum hazards
5.	Explain memory organization, I/O organization and its impact on computer cost /performance.

Module I: Basic Structures of Computers [5 Lectures]

Introduction to Digital Logic, Basic Structure of Computers: Computer Types, Functional Units, Input Unit, Memory Unit, Arithmetic and Logic Unit, Output Unit, Control Unit, Basic Operational Concepts: Fixed and floating point Representation and Arithmetic Operations, Performance, Historical Perspective.

Module II: Instruction Set Architecture [5 Lectures]

Memory Locations and Addresses: Byte Addressability, Big-Endian and Little-Endian Assignments, Word Alignment, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Subroutines, Additional Instructions, Dealing with 32-Bit Immediate Values.

Module III: Basic Processing Unit & Pipelining [10 Lectures]

Some Fundamental Concepts, Instruction Execution, Hardware Components, Instruction Fetch and Execution Steps, Control Signals, Hardwired Control, CISC-Style Processors. Pipelining**:** Basic Concept, Pipeline Organization, Pipelining Issues, Data Dependencies, Memory Delays, Branch Delays, Pipeline Performance Evaluation.

Module IV: Memory Organization [10 Lectures]

Basic Concepts, Semiconductor RAM Memories, Read-only Memories, Direct Memory Access, Memory Hierarchy, Cache Memories, Performance Considerations, Virtual Memory, Memory Management Requirements, Secondary Storage.

Module V: Input Output & Parallel Processing [10 Lectures]

Basic Input Output Accessing I/O Devices, Interrupts, Input Output Organization Bus Structure, Bus Operation, Arbitration, Interface, Interconnection Standards.Parallel Processing, Hardware Multithreading, Vector (SIMD) Processing, Shared-Memory Multiprocessors, Cache Coherence, Message-Passing Multicomputers, Parallel Programming for Multiprocessors, Performance Modeling.

Text Book:

1. Patterson David A., Hennessy John L., Computer Organization and Design: The Hardware / Software Interface, 5th Edition, Elsevier.

Reference Books:

1. Hamachar Carl et. al, Computer Organization and Embedded Systems, 6th Edition, McGraw Hill.
2. Mano M. Morris, Computer System Architecture, Revised 3rd Edition, Pearson.

SEMESTER III LABORATORIES

Course code**: EC204**
Course title**: Digital System design Lab**
Pre-requisite(s)**: EC101 Basics of Electronics & Communication Engineering**
Co- requisite(s):
Credits: L:0 T:0 P:3 C:1.5
Class schedule per week**: 03**
Class**: B. Tech**

Course Outcomes

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

CO1	Describe the knowledge of basic logic gates and their design using universal gates.
CO2	Demonstrate the working of combinational and sequential circuits.
CO3	Integrate and experiment with controlled digital circuits.
CO4	Appraise combinational/sequential circuits and memories.
CO5	Schematize, simulate and implement combinational and sequential circuits to solve real world problems using VHDL systems.

List of experiments:

1. Design and implement a controlled CMOS Inverter.
2. To study and verify the truth table of NAND and EX-OR gate using IC 7400.
3. Design and implement SEVEN segment display unit.
4. Design and verify half adder and full Adder circuits using gates and IC 7483.
5. Design and implement a 3:8 Decoder.
6. Design and implement 8:3 priority encoder.
7. Design a 4 bit magnitude comparator using combinational circuits.
8. Design and implement 8:1 multiplexer and 1:4 demultiplexer.
9. Design ALU with functions of ADD, SUB, INVERT, OR, AND. XOR, INC, DEC and CMP.
10. Design and verify decade Counter.
11. Design a ROM (8X4) using decoder, gates and diodes.
12. Design of pre settable up/down counter.

Implement all the above experiments using VHDL platform and verify.

Books recommended:

Textbooks:

1. "Digital Design", Morris Mano and Michael D. Ciletti ,5th edition PHI
2. "Digital System Design using VHDL", Charles H Roth, Thomson Learning

Reference books:

1. Digital computer Electronics AP Malvino, 3rd Edition Mc Graw Hill

Course code: **CS232**
Course title: **Data Structures Lab**
Pre-requisite(s):
Co- requisite(s): **Data Structures**
Credits: **L: 0 T: 0 P: 3**
Class schedule per week: **3**
Class: B. Tech
Branch: CSE/AIML

Course Outcomes

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

1.	Be able to design and analyze the time and space efficiency of the data structure
2.	Analyze run-time execution of previous learned sorting methods, including selection, merge sort, heap sort and Quick sort
3.	Have practical knowledge on the applications of data structures
4.	Be capable to identity the appropriate data structure for given problem

SYLLABUS

1. Program to Find the Number of Elements in an Array
2. Develop and Implement a menu driven program in C for the following Array operations
    a. Creating Array of N Integer elements.
    b. Display of Array elements with suitable headings.
    c. Inserting an element (ELEM) at a given valid position (POS).
    d. Deleting an element at a given valid position (POS).
    e. Exit
3. Programs for Stack, Queues and Circular Queues using Arrays
4. Program to convert an Infix Expression into Postfix and Postfix Evaluation
5. Program to implement stack using arrays
6. Program to implement stack using linked list
7. Program to implement multiple stack in a single array
8. Program to convert infix notation to postfix notation using stacks
9. Program to implement queue using arrays
10. Program to implement queue using pointers
11. Program to reverse elements in a queue
12. Program to implement circular queue using arrays
13. Program to create add remove & display element from single linked list
14. Program to create add remove & display element from double linked list
15. Program to count number of nodes in linear linked list
16. Program to create add remove & display element from circular linked list
17. Programs to implement stack & queues using linked representation
18. Program to concatenate two linear linked lists
19. Program to accept a singly linked list of integers & sort the list in ascending order.
20. Program to reverse linked list
21. Program to represent polynomial using linked list

22. Program to add two polynomials using linked list
23. Program for the creation of binary tree, provide insertion & deletion in c
24. Program for pre-order, post-order & in-order traversals of a binary tree using non reccursive.
25. Program to count no, of leaves of binary tree
26. Program for implementation of B-tree (insertion & deletion)
27. Program for implementation of multi-way tree in c
28. Program for implementation of AVL tree
29. Program to implement bubble sort program using arrays
30. Program to implement merge sort using arrays
31. Program to implement selection sort program using arrays
32. Program to implement insertion sort program using arrays
33. Program to implement topological sort using arrays
34. Program to implement heap sort using arrays
35. Program to implement heap sort using pointers
36. Program to implement bubble sort program using pointers
37. Program to implement linear search using pointers
38. Program to implement binary search using pointers
39. Program to implement linear search using arrays
40. Program to implement binary search using arrays

Text books:

1. Baluja G S, "Data Structure through C", Ganpat Rai Publication, New Delhi, 2015.
2. Pai G A V, "Data Structures and Algorithms: Concepts, Techniques and Applications", 2^{nd}Edn, Tata McGraw-Hill, 2008.
3. Horowitz E., Sahni S., Susan A., "Fundamentals of Data Structures in C", 2^{nd} Edition, University Press, 2010.

Reference books:

1. Tremblay J. P., Sorenson P. G, "An Introduction to Data Structures with Applications", 2nd Edn, McGraw-Hill, Inc. New York, NY, USA.
2. Lipschutz Seymour, "Data Structures", 6th Edn, 9th Reprint 2008, Tata McGraw-Hill.
3. Drozdek Adam, "Data Structures and Algorithms in C++", Thomson Learning, New Delhi – 2007.
4. Feller J., Fitzgerald B., "Understanding Open Source Software Development", Pearson Education Ltd. New Delhi

Course code: **CS234**
Course title: **OOPDP Lab**
Pre-requisite(s):
Co- requisite(s): Object Oriented Programming & Design Principles
Credits: **L: 0 T: 0 P: 3**
Class schedule per week: **3**
Class: **B. Tech**
Branch: **CSE/AIML**

Course Outcomes

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

CO1	Work in any object oriented environment and program using those features.
CO2	Student will have hands on experience with all basic concepts of Java programming
CO3	Analyse the design pattern of the given problem and further solve with less complexity.
CO4	Use his/her programming skills to resolve the issues coming while programming for bigger problems.
CO5	Work in industry environment with good enough knowledge about Java and OOPs.

Syllabus

List of Programs as Assignments:

1. **Lab Assignment No: 1**

Objective: To understand and Implement basic java program concepts using Scanner class.

Q1. Take input from user a character variable in a program and if the value is alphabet then print "Alphabet" if it's a number then print "Digit" and for other characters print "Special Character"

Q2. Write a program to add all the values in a given number and check if the sum is prime number or not. Ex: 1234->10, not prime.

2. **Lab Assignment No: 2**

Objective: To Understand and Implement the concept of arrays in java

Q1. Write a program to find the largest 2 numbers and the smallest 2 numbers in the array initialized by the user.

Q2. Write a program to print the element of an array that has occurred the highest number of times Eg) Array -> 10,20,10,30,40,100,99 O/P:10

3. **<u>Lab Assignment No: 3</u>**

Objective: To Understand and Implement the concept of 2-D arrays in java.

Q1.Write a program to reverse the elements of a given 2*2 array. Four integer numbers needs to be passed as Command Line arguments

Eg: C:\>java Sample 1 2 3 4
O/P Expected :
The given array is : 1 2
3 4
The reverse of the array is : 4 3
2 1

Q2. Write a program to find greatest number in a 3*3 array. The program is supposed to receive 9 integer numbers as command line arguments.

4. **<u>Lab Assignment No: 4</u>**

Objective: To Understand and Implement the concept of classes and Constructors

Q1. Create a class Box that uses a parameterized constructor to initialize the dimensions of a box.(dimensions are width, height, depth of double type). The class should have a method that calculates and returns the volume of the box . Obtain an object and print the corresponding volume in main() function.

Q2. Write a program in Java with class Rectangle with the data fields width, length, area and color. The length, width and area are of double type and color is of string type. The methods are set_length() , set_width() , set_color(), and find_area(). Create two object ofRectangle and compare their area and color. If area and color same for the objects then display "Matching Rectangles" otherwise display "Non Matching Rectangle".

5. **<u>Lab Assignment No: 5</u>**

Objective: To Understand and Implement the concept of Inheritance

Q1.Create a class named 'Animal' which includes methods like eat() and sleep(). Create a child class of Animal named 'Bird' and override the parent class methods. Add a new method named fly(). Create an instance of Animal class and invoke the eat and sleep methods using this object.Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.

Q2. A HighSchool application has two classes: the Person superclass and the Student subclass. Using inheritance, in this lab you will create two new classes, Teacher and CollegeStudent. A Teacher will be like Person but will have additional properties such as salary (the amount the teacher earns) and subject (e.g. "Computer Science", "Chemistry", "English", "Other"). The CollegeStudent class will extend the Student class by adding a year (current level in college) and major (e.g. "Electrical Engineering", "Communications", "Undeclared").

6. **<u>Lab Assignment No: 6</u>**

Objective: To Understand and Implement the concept of Overloading and Overriding

Q1.Create a class Account with two overloaded constructors. First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, address, type of account and current balance. Account class is having methods Deposit(), WithDraw(), and GetBalance().Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them.

Q2. Create a base class Fruit which has name ,taste and size as its attributes. A method called eat() is created which describes the name of the fruit and its taste. Inherit the same in 2 other class Apple and Orange and override the eat() method to represent each fruit taste.

7. **Lab Assignment No: 7**

Objective: To Understand and Implement String class in Java

Q1. Reverse the string but not the words. Eg. I/P: Birla institute of technology

O/P: technology of institute birla.

Q2. Find out and print the maximum possible palindrome in a given string. Eg:

I/P: nononsene O/P: nonon

Q3.Given a string and a non-empty word string, return a string made of each char just before and just after every appearance of the word in the string. Ignore cases where there is no char before or after the word, and a char may be included twice if it is between two words.

If inputs are "abcXY123XYijk" and "XY", output should be "c13i".

If inputs are "XY123XY" and "XY", output should be "13".

8. **Lab Assignment No: 8**

Objective: To Understand and Implement the concept of Abstract classes and Interfaces

Q1. Create an abstract class Compartment to represent a rail coach. Provide an abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment. Create a class TestCompartment. Write main function to do the following: Declare an array of Compartment of size 10. Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behavior of the notice method.

Q2.Write a program in java which implement interface Student which has two methods Display_Grade and Attendance for PG_Students and UG_Students (PG_Students and UG_Students are two different classes for Post Graduate and Under Graduate Students respectively).

9. **Lab Assignment No: 9**

Objective: To Understand and Implement Exception handling in java

Q1. Write a program in Java to display name and roll number of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsException, so that any such problem does not cause illegal termination of program.

Q2. Write a program to accept name and age of a person from the command prompt(passed as arguments when you execute the class) and ensure that the age entered is >=18 and < 60. Display proper error messages. The program must exit gracefully after displaying the error message in case the arguments passed are not proper. (Hint : Create a user defined exception class for handling errors.)

10. **Lab Assignment No: 10**

Objective: To Understand and Implement File Handling and multithreading in java

Q1. Write a program to count the number of times a character appears in the Fileand also copy from one file to another. (Case insensitive... 'a' and 'A' are considered to be the same)

Q2. 1. Create class of SalesPersons as a thread that will display fives sales persons name. 2. Create a class as Days as other Thread that has array of seven days.

3. Call the instance of SalesPersons in Days and start both the threads 4. suspendSalesPersons on Sunday and resume on wednesday Note: use suspend, resume methods from thread

Q3. Create two threads, one thread to display all even numbers between 1 & 20, another to display odd numbers between 1 & 20. Note: Display all even numbers followed by odd numbers Hint: use join

11. Lab Assignment No: 11

Objective: To Understand and Implement Applets, AWT and Swings

Q1. Program to create a calculator with the help of AWT packages in Java.

Q2. Program to create a unit converter using Swings in Java.

Q3. APPLETS

a) Working with Frames and various controls.

b) Working with Dialogs and Menus.

c) Working with Panel and Layout.

d) Incorporating Graphics.

e) Working with colors and fonts

TEXT BOOKS

1. Krishna P. R., Object Oriented Programming through JAVA, 1st Edition, Universities Press, 2008.
2. Patrick Naghton& H. Schildt – The Complete Reference Java 2, Tata McGraw Hill Publication, New Delhi.
3. Dietel,Dietel - Java How to program , 7th edition; Pearson Education , New Delhi.

REFERENCE BOOKS

1. C. Horstmann,G. Cornell - Core Java 2 Vol I & Vol II ; Pearson Education , New Delhi.
2. Balagurusamy -Programming in Java, 2nd Edition; Tata McGraw Hill Publication; New Delhi.

SEMESTER IV COURSE INFORMATION SHEET

COURSE INFORMATION SHEET

Course Code: AI201

Course Title: Probability and Statistical Analysis

Pre-requisite(s):

Co- requisite(s):

Credits: L: 3 T: 0 P: 0

Class Schedule Per Week: 3

Class: B. Tech

Semester / Level: IV/2

Branch: AIML

Course Objectives:

1. Apply descriptive and inferential statistical techniques to summarize, visualize, and interpret data effectively.
2. Explain the basic principles of probability theory, including conditional probability, bayes theorem, probability distributions and random variables.
3. Compute probabilities using discrete and continuous probability distributions of random variables, including Binomial, Poisson, Uniform, Exponential and Normal distributions.
4. Utilize probabilistic models and statistical methods to solve problems in fields such as engineering,business and other areas.
5. Formulate and test statistical hypotheses using confidence intervals, p-values, and various statistical tests.

Course Outcomes:

After the completion of this course, students will be:

1. Able to describe and summarize large real-life datasets.
2. Able to compute the probability of complex events.
3. Able to describe the characteristic properties of different types of random variables.
4. Able to estimate distribution parameters for real life samples.
5. Able to perform hypothesis testing on datasets and interpret the results.

Syllabus:

Module 1: Descriptive Statistics **[8 Lectures]**
Introduction, Describing Datasets, Summarizing Datasets, Chebyshev's Inequality, Normal Datasets, Paired Datasets and Correlation coefficient, Lorenz Curve and Gini Index.

Module 2: Elements of Probability **[8 Lectures]**

Basic concepts, Conditional Probability, Baye's formula, Independent Events, Random variables, Types of Random variables, jointly distributed random variables, Expectation and its properties, Variance and Covariance

Module 3: Special Random Variables **[8 Lectures]**

The Bernoulli and binomial random variable, The Poisson random variable, The uniform random variable, The normal random variable, Exponential Random variables, The gamma distribution, The Chi-square, t-distribution, and F-distribution.

Module 4: Parameter Estimation **[8 Lectures]**

Maximum Likelihood Estimators, Interval Estimates, Estimating the difference in means of two normal distributions.

Module 5: Hypothesis Testing **[8 Lectures]**

Significance Levels, Tests about mean of a normal population, Testing the equality of means of two normal populations. Tests concerning the variance of a normal population.

Textbooks:

1. Ross Sheldon M., Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Academic Press, 2021

Reference Books:

1. Rohatagi V. K., Saleh A. K., An Introduction to Probability and Statistics, 3rd Edition, Wiley, 2015

Course Outcomes (COs) Attainment Assessment Tools andEvaluation Procedure:

Direct Assessment Tool	**% Contribution During Co – Assessment**
Mid Semester Examination	25
End Semester Examination	50
Quiz 1 + Quiz 2	10+10
Assignment	5

Assessment Components	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**
Mid Semester Examination	√	√	√		
End Semester Examination	√	√	√	√	√
Quiz 1 + Quiz 2	√	√	√	√	
Assignment				√	

Indirect Assessment:

1. Student Feedback on Course Outcomes (COs)

Mapping of Course Outcomes (COs) onto Program Outcomes (POs) and Program Specific Outcomes (PSOs):

Course Outcomes (COs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	3	3	1	3	2	1	1	1	2	2	3	2	1	3	1
CO2	3	3	3	3	2	1	1	1	2	2	2	1	3	3	1
CO3	2	1	3	1	1	1	1	1	1	1	1	1	3	3	1
CO4	2	1	3	1	1	1	1	1	1	1	1	1	1	1	1
CO5	2	1	3	1	1	1	1	1	1	3	1	1	3	3	1

Course code: AI203
Course title: Mathematics for Data Science
Pre-requisite(s): Maths-I. Maths-II
Co- requisite(s):
Credits: L:3 T:0 P:0
Class schedule per week: 3
Class: B. Tech
Semester / Level: IV
Branch: CSE/IT/AIML

Course Outcomes:
On completion of the course students shall be:
1) Able to formulate practical problems as graphs and use graph theoretic methods on them.
2) Able to model transfer of information and state using Markov chains.
3) Able to reduce the dimensionality of large datasets and understand the challenges in doing so.
4) Able to interpolate missing data both for linear and non linear cases over multiple dimensions.
5) Apply concepts of optimization to machine learning problems

Module 1: Graph Theory	**Lectures: 8**
Basic concepts and terminology, Adjacency matrix – representation and uses, Eigen values and Eigen vectors – importance and applications	

Module 2: Stochastic Processes	**Lectures: 8**
Markov chain basics, Hidden Markov models, - the Likelihood problem, the decoding problem, the Learning problem.	

Module 3: SVD & PCA	**Lectures: 8**
Vector and inner product spaces. Singular values, Singular Value Decomposition, Compression of data using PCA, relation of PCA with co-variance and correlation, Linear Discriminant Analysis	

Module 4: Interpolation	**Lectures: 8**
Lagrange interpolation, Orthogonal family of polynomials, Newton divided difference methods – use of Vandermonde matrix, Chebyshev interpolation, Hermite regression, Least square regression	

Module 5: Optimization and Learning Techniques	**Lectures: 8**
Basic probability theory, Introduction to matrix calculus – Matrix differentiation, Matrix integration, Maximum Likelihood Estimation, Gradient Descent,	

Textbook:
1) Arangala C., “Linear Algebra with Machine Learning and Data”, CRC Press, 1st Edition, 2023

Reference Books:
1) Carter N. “Data Science for Mathematicians”, CRC Press, 1st Edition, 2021
2) Strang G. “Introduction to Linear Algebra”, 5th Edition, Cambridge Press, 2016

Course code: **CS241**
Course title: **Design and Analysis of Algorithm**
Pre-requisite(s): **Data Structures**
Co- requisite(s): **Algorithms Lab**
Credits: **L: 3 T: 0 P: 0**
Class schedule per week: **3**
Class: **B. Tech**
Branch**: AI/ML**

Course Outcomes

After the completion of this course, students will be:

1.	Define the concepts and mathematical foundation for analysis of algorithms.
2.	Explain different standard algorithm design techniques, namely, divide & conquer, greedy, dynamic programming, backtracking and branch & bound.
3.	Demonstrate standard algorithms for fundamental problems in Computer Science.
4.	Design algorithms for a given problem using standard algorithm design techniques.
5.	Analyze and compare the efficiency of various algorithms of a given problem.

Module 1:Algorithm & Complexity [8 Lectures]
Introduction, Algorithm Complexity and various cases using Insertion Sort, Asymptotic Notations, Time complexity of Recursive Algorithm, Solving Recurrences using Iterative, Recursion Tree and Master Theorem.

Module 2:Divide & Conquer [8 Lectures]
Discussion of basic approach using Binary Search, Merge Sort , Quick Sort , Selection in Expected linear time, Maximum Subarray , Matrix Multiplication , Introduction of Transform and Conquer and AVL Tree .

Module 3: Dynamic Programming [8 Lectures]
Introduction and Approach, Rod Cutting, LCS, Optimal BST, Transitive closure and All-pair Shortest Path, Travelling Salesperson Problem.

Module 4: Greedy & Other Design Approaches [8 Lectures]
Introduction to greedy using fractional knapsack, Huffman Code, Minimum Spanning Tree – Prim and Kruskal, Single Source Shortest Path Dijkstra's and Bellman-Ford, Introduction to Backtracking using N-Queens problem, Introduction to Branch and Bound using Assignment Problem or TSP.

Module 5: NP Completeness and Other Advanced Topics [8 Lectures]
Non-deterministic algorithms – searching and sorting, Class P and NP, Decision and Optimization problem, Reduction and NPC and NPH, NP Completeness proof for: SAT, Max-Clique, Vertex Cover, Introduction to Randomized Algorithms, Introduction to Approximation Algorithms.

Text Books:

1. Cormen Thomas H. et al., Introduction to Algorithms. 3rd Edition, PHI Learning, latest edition.

Reference Books:

1 Horowitz E., Sahani, Fundamentals of Computer Algorithms, Galgotia Publication Pvt. Ltd.

2 Dave and Dave, Design and Analysis of Algorithms, 2nd Edition, Pearson.

3 Goodrich, Tamassia. Algorithm Design. Wiley.

Course code: AI205
Course title: Introduction to AI
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:1 P:0
Class schedule per week: 4
Class: B. Tech
Semester / Level: IV/ II
Branch: CSE/IT/AIML

Course Objectives

1) To familiarize with knowledge representation concepts.
2) To understand problem formulation and choice of informed and uninformed search.
3) To understand Classical and Heuristic methods and solve game problems.
4) To learn reinforcement learning.
5) To learn how to solve real life probabilistic problems

Course Outcomes:
On completion of the course students shall be:

1) Able to formulate propositional logic for real life problems and present formal proofs for standard problems.
2) Solve problems using First Order Predicate Logic and work with automatic Theorem Provers.
3) Perform Search using Classical and Heuristic methods and solve game problems.
4) Design entropy-based solutions for real life probabilistic problems.
5) Understand the basics of reinforcement systems and their operating principals.

Module 1: Introduction and Propositional Logic	**Lectures: 8**
What Is Artificial Intelligence? Agents, Knowledge-Based Systems, Propositional Logic – Syntax, Semantics, Proof Systems, Resolution, Horn Clauses, Computability and Complexity	

Module 2: First Order Predicate Logic	**Lectures: 8**
Syntax, Semantics, Quantifiers and Normal Forms, Proof Calculi, Resolution, Automated Theorem Provers, Applications and Limitations	

Module 3: Search Games and Problem Solving	**Lectures: 8**
Introduction, Uninformed Search, Heuristic Search, Games with Opponents, Heuristic Evaluation functions	

Module 4: Reasoning with Uncertainity	**Lectures: 8**
Computing with Probabilities, The principal of Maximum Entropy, Reasoning with Bayesian Networks	

Module 5: An Introduction to Reinforcement Learning	Lectures: 8
Definitions, Uninformed combinatorial search, Value iteration and Dynamic Programming, Q-Learning, Exploration and Exploitation, Approximation, Generalization and Convergence, the Curse of Dimensionality	

Textbook:

1) Ertel W.., "Introduction to Artificial Intelligence", UTiCS Springer, 2nd Edition, 2017
2) Akerkar R. "Introduction to Artificial Intelligence", 2nd Edition, PHI Press, 2014

Reference Books:

1) Russell S., Norvig P. "Artificial Intelligence: A Modern Approach", Pearson Publications, 4tht Edition, 2022
2) Akerkar R. "Introduction to Artificial Intelligence", 2nd Edition, PHI Press, 2014

Mapping CourseOutcomesonto ProgramOutcomes

CO PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO12	PSO 1	PSO2	PSO 3
CO1	High	Medium	Medium	Medium	Medium	High	Low	Low	Medium	Medium	No	Low	High	Medium	Medium
CO2	High	High	High	High	High	High	Low	Low	Medium	Medium	No	Low	High	High	Medium
CO3	High	High	High	High	High	Low	Low	Low	Medium	Medium	No	Medium	High	High	High
CO4	High	High	High	High	High	Low	Low	Low	Medium	Medium	No	Medium	High	High	High
CO5	High	Medium	Medium	High	High	Medium	Low	Low	Medium	Medium	No	Low	Medium	Medium	Medium

SEMESTER IV LABORATORIES

Course Code: AI 202
Course Title: IT/ Systems Workshop (LEX, YACC)
Pre-requisite(s) : CS102 : PPS Lab & CS232 : Data Structures Lab
Co- requisite(s) : CS 240 (Shell and Kernel Programming Lab)
Credits: L: 0 T: 0 P: 2
Class schedule per week: 2
Class : B. Tech
Semester / Level: IV/II
Branch: AIML
Course Objectives: This course enables the students to

1. Learn the formalism of regular expression (regex) to represent patterns of interest (tokens) required in various applications.
2. Learn how to convert a regex automatically into a deterministic finite automaton using the tool,

Lex.

3. Acquire skills in writing context free grammar (CFG) to capture the syntactic structure of objects of interest.
4. Learn how to automatically generate a syntax analyser (parser) from a context free grammar using the software tool, YACC.
5. Given a pattern recognition problem, identify the pattern features that can be specified by regular expressions and other features that can be captured by a context free grammar. Use tools to generate a software for solving the problem and test its validity.

Course Outcomes: After the completion of this course, students will be able to
CO1 Write regular expressions for various strings of interest, such as, name, number in a given base, real number, date, keywords, etc.
CO2 Write regex for a pattern in the notation of Lex, attach actions in C language, and use the definition sections of Lex to generate a C program that becomes the pattern recognizer.
CO3 Given a set of strings with a desired structure, to write a context free grammar (CFG), first in a formal notation, then in the notation of YACC; generate a syntax analyser.
CO4 Communicate between the pattern recognizer generated by lex and the syntax analyser generated by yacc in order to create a software that recognizes strings belonging to a context free language. Acquire skills to interpret the outputs generated by Lex and YACC.
CO5 Add semantic actions to the CFG rules of YACC to perform analysis of the input string and/or undertake transformations as appropriate to the problem statement

SYLLABUS

1. Experimentation with the use of a production quality open source compiler, such as gcc (gnu C Compiler), to get insight about its working. Familiarity with the internal dumps provided by the compiler. Review of useful shell commands, working with an editor, such as vi, and working knowledge of Linux operating system.
2. Introduce the basic theoretical concepts related to regular expressions (regexes), such as finite alphabet, strings over the alphabet, and languages. Definition of regular expression (regex), regex operators and their properties, language denoted by a regex.
3. Introduction to Lexical analyser generator tool, lex. Structure of a lex script. Write regex in Lex notation and conduct experiments with lex to convert regular expressions to a minimal deterministic finite automation (DFA which is also a pattern recognizer). Test the

performance of the recognizer over different input strings, few belonging to the language and few not in the language of regex.

4. Introduce the basic theoretical concepts related to a context free grammar (CFG) and LALR(1) parsing, such as terminals, non-terminals, production rules, derivations, parse tree, language corresponding to CFG, LR(0) item, parser actions - shift, reduce, accept and error, LALR(1) parsing table and LALR(1) automaton. Write grammars for objects with simple structure.

5. Introduction to parser generator tool, YACC (Yet Another Compiler Compiler). Structure of an yacc script. Write a CFG in yacc notation, conduct experiments with yacc to generate a parser from the given CFG. Understand and interpret the generator parser. Test the performance of the parser over different input strings, few belonging to the language of the grammar and few strings which are not.

6. Given an application, use Lex and yacc to generate a software that recognize objects with the desired structure and produce output after performing appropriate transformations as required in the application.

Text Books:

1. Lex and Yacc, John R. Levine, Tony Mason, Doug Brown [1992], O'Reilly & Associates, 2nd Edition, (ebook available)

References :

2. Lex and Yacc Tutorial, Tom Niemann, epaperpress.com

3. Johnson, Stephen C. [1975]. Yacc: Yet Another Compiler Compiler. Computing Science Technical Report No. 32, Bell Laboratories, Murray hill, New Jersey. A PDF version is available at e Paper Press.

4. Lesk, M. E. and E. Schmidt [1975]. Lex – A Lexical Analyzer Generator. Computing Science Technical Report No. 39, Bell Laboratories, Murray Hill, New Jersey. A PDF version is available e-Paper Press.

Course code: AI204
Course title: Mathematics for Data Science Lab.
Pre-requisite(s):
Co- requisite(s): Mathematics for Data Science
Credits: L: T: P: 3
Class schedule per week: 3
Class: B. Tech
Semester / Level: IVth
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	Able to convert real life problems into graphs.
2.	Able to solve simple system of equations
3.	Fit straight lines and curves to datasets
4.	Compute eigen values for matrices and factorize large matrices.
5.	Detect optimum points for fitness functions.

Module 1: Graph Algorithms **[8 Lectures]**
Representation of graphs using adjacency matrix, Traversal of Graphs, Shortest path algorithms, Detecting spanning trees

Module 2: System of Equations **[8 Lectures]**
Representing system of equations, Gauss Elimination methods, LU Decomposition, Matrix inversion.

Module 3: Interpolation and Curve Fitting **[8 Lectures]**
Fitting straight lines, Polynomial Interpolation, Interpolation with Cubic spline.

Module 4: Matrix Methods **[8 Lectures]**
Reduction using PCA, Matrix Factorization, Computing Eigen values and vectors.

Module 5: Optimization and Statistics **[8 Lectures]**
Finding maxima and minima using Hill climbing techniques, Gradient descent. Generating data for standard distributions.

SEMESTER V COURSE INFORMATION SHEET

COURSE INFORMATION SHEET

Course Code: IT333
Course Title: Data Communication & Computer Networks (DCCN)
Pre-requisite(s): Operating System, Digital System Design
Co-requisite(s):
Credits: L:3 T:1 P:0
Class Schedule per week: 4
Class: B.Tech.
Semester/Level: V
Branch: CS/AI-ML

Course Objectives:

This course enables the students to:

1.	Study the basic communication model, types of networks, applications and layered protocol architecture.
2.	Understand characteristics of transmission media, types of impairments, error detection and correction methods.
3.	Understand digital & analog transmission systems, multiplexing, digital encoding and modulation techniques.
4.	Understand basics of local area networking, error and flow control mechanisms and shared channel access mechanisms.
5.	Understand internetworking concepts, IP addressing & routing techniques, and transport protocols.

Course Outcomes

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

1.	Comprehend communication protocol architecture, identify network types and map functions to real-life applications.
2.	Examine signal impairments, determine channel capacity, analyze transmission characteristics of different mediums.
3.	Simulate analog and digital signaling methods, multiplexing & encoding techniques and compare efficiency.
4.	Analyze local area networking implementations, present error and flow control solutions.
5.	Experiment with internetworking protocols, analyze and compare their functions and evaluate efficiency.

Syllabus

Module I

Basics of Data Communications and Networking: Overview: Communication Model, Types of Networks, Internet, Protocol Architecture Standards, TCP/IP Architecture, Internet Applications.

Module II

Data Transmission: Analog and Digital Transmission System, Transmission Impairments, Channel Capacity, Guided Transmission Mediums, Wireless Transmission and Propagation.

Signal encoding: Digital Signaling and Analog Signaling, Encoding Techniques, Modulation Techniques.

Module III

Multiplexing: Frequency Division Multiplexing and Synchronous Time Division Multiplexing
Error Handling: Types of Errors, Error Detection and Correction Mechanisms.
Data Link Control Protocols: Data Link Error and Flow Control, HDLC Protocol.

Module IV

Local Area Networking: Topologies, Protocol Architecture, Virtual LANs, Multiple Channel Access Mechanisms, Traditional and High-Speed Ethernet, Wireless LANs, IEEE 802.11 Overview.

Module V

Internetworking: Internetworking Principles, Circuit Switching and Packet Switching, Internet Protocol, DHCP Protocol, Transport Protocols, Routing in Packet Switched Networks, Quality of Service Parameters, Integrated Services, Differentiated Services, MPLS.

Textbook:

Stallings W., Data and Computer Communications, 10^{th} Edition, Pearson Education, PHI, New Delhi, 2017.

Reference Book:

Forouzan B. A., Data Communications and Networking, 6^{th} Edition, TMH, New Delhi, 2022.

Gaps in the syllabus (to meet Industry/Profession requirements):N/A

POs met through Gaps in the Syllabus:N/A

Topics beyond syllabus/Advanced topics/Design:N/A

POs met through Topics beyond syllabus/Advanced topics/Design:N/A

Course Outcome (CO) Attainment Assessment Tools and Evaluation Procedure

Direct Assessment

Assessment Tools	% Contribution during CO Assessment
Continuous Internal Assessment	50
Semester End Examination	50

Continuous Internal Assessment	% Distribution
Mid semester examination	25
Two quizzes	20(2×10)
Teacher's Assessment	5

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	√	√	√	√	√
Semester End Examination	√	√	√	√	√

Indirect Assessment

1. Student Feedback on Faculty
2. Student Feedback on Course

Course Delivery Methods	
CD1 CD2	Lecture by use of boards/LCD projectors/OHP projectors
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini Projects
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course Outcomes onto Program Outcomes

Course Outcome	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CO1	3	3	3	3	2	1		1	1	2	2	3	3	2	2
CO2	3	3	3	2	2	1	1		1	2	1	3	3	3	3
CO3	3	3	3	3	2	2	1	1	1	2	1	3	3	3	3
CO4	3	3	3	3	3	2	2	1	2	3	2	3	3	3	3
CO5	3	3	3	3	3	2	2	1	2	3	2	3	3	3	3

Correlation Levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

MAPPING BETWEEN COURSE OUTCOMES AND COURSE DELIVERY METHOD

Course Outcomes	Course Delivery Method
CO1	CD1,CD6
CO2	CD1, CD6,CD7
CO3	CD1, CD2, CD3,CD6,CD7
CO4	CD1, CD3,CD6,CD7
CO5	CD1,CD2,CD3,CD4,CD5,CD7

Course code: **CS237**
Course title: **Database Management System (DBMS)**
Pre-requisite(s): **Data Structures.**
Co- requisite(s):
Credits: **L:3 T:0 P:0**
Class schedule per week: **3**
Class: **B. Tech**
Branch**: AI/ML**

Course Outcomes

After the completion of this course, students will be:

1.	Analyze data organization requirements and their inter relationships.
2.	Illustrate the features of data models and their application for storing data.
3.	Design queries to maintain and retrieve useful information from the databases created.
4.	Analyze the physical database design with respect to their expected performance using normalization and query processing.
5.	Examine the best practices according to concepts of indexing, transaction control and concurrency maintenance

Module 1: Database Design & Entity-Relational Model [8 Lectures]
Purpose of Database System; View of Data, Database Languages, Transaction Management, Database architecture, Database Users and Administrator, Types of database System, Overview of design process, E-R model, Constraints, E–R Diagram, E-R Diagram issues, Weak Entity Sets, Extended E – R Features, Reduction to E–R Schemas.

Module 2:Relational Model [8 Lectures]
Structure of Relational Database, Codd's Rules, Fundamental Relational Algebra Operations, Additional Relational Algebra Operations, Extended Relational Algebra Operations, Data definition, Basic structure of SQL queries, Set Operations, Aggregate Functions, Null Values, Nested Sub Queries, complex queries, views, modification of database, Joined relations, SQL data types & schemas, Integrity constraints, authorization, Embedded SQL, Triggers.

Module 3: Relational Database Design [8 Lectures]
Functional dependency, Decomposition, Normalization, First normal form, Second normal form, Third normal form, BCNF, Multivalued dependencies and Fourth normal form, Join dependencies and Fifth normal form, DKNF.

Module 4: Indexing & Hashing [8 Lectures]
Ordered Indices, B+ Tree index files, B-Tree index files, Multiple key access Static hashing, Dynamic Hashing, Comparison of ordered indexing and hashing, Index definition in SQL.
Query Processing - Measure of Query Cost, Selection Operation, Evaluation of Expressions.

Module 5: Transaction & Concurrency Control **[8 Lectures]**
Transaction Concepts & ACID Properties, Transaction States, Implementation of Atomicity & Durability, Concurrent Executions, Serializability& Its Testing, Recoverability, Lock-Based protocols, Validation based protocol, Multiple Granularity, Multiversion Schemes, Deadlock Handling.

Text Books:

1. Silberschatz A. et.al, Database System Concepts, 6th Edition, Tata Mc-Graw Hill, New Delhi, 2011.

Reference Books:

1. Elmasri R., Fundamentals of Database Systems, 7th Edition, Pearson Education, New Delhi, 2016.
2. Ullman Jeffrey D et.al., A First course in Database Systems, 3rd Edition, Pearson Education, New Delhi-2014.

COURSE INFORMATION SHEET
Course code: AI301 **Course title:** Supervised Learning **Pre-requisite(s):** **Co- requisite(s):** **Credits:** L:3 T:0 P:0 **Class schedule per week**: 3 **Class:** B. Tech **Semester / Level**: V/ III **Branch**: AIML

Course Objectives

1) To familiarize with the core concepts of supervised learning.
2) To understand the working of a simple neural network with backpropagation.
3) To understand the technique of SVMs and decide the proper kernel for a task.
4) To learn about tree based classifiers and demonstrate the splitting criteria.
5) To learn how Ensemble works.

Course Outcomes:

On completion of the course students shall be:

1) Able to demonstrate with examples the core concepts of supervised learning.
2) Able to articulate the working of a simple neural network with backpropagation.
3) Able to perform classification using SVMs and decide the proper kernel for a task.
4) Able to build tree based classifiers and demonstrate the splitting criteria.
5) Able to demonstrate how Ensemble learning helps improve classifier performance.

Module 1: Preliminaries	**Lectures: 8**
Terminology, Regression, Classification, Weight Space, Curse of dimensionality, Overfitting, Training, testing and Validation Sets, Confusion Matrix, Accuracy Metrics, ROC Curve, Unbalanced datasets, Precision, Probability, Naïve Bayes' classification, Basic Statistics – Average, Variance and Covariance, Gaussian.	

Module 2: Neural Networks, Linear Separability and Multi-Layer Perceptron	**Lectures: 12**
Hebb's rule, McCullogh and Pitts Neurons, Limitations of the basic neurons, the Perceptron, the concept of Linear Separability, Linear Regression, The concept of Bias, Backpropagation and its working, Practical aspe of learning – Amount of training data, number of hidden layers, when to stop learning, Deriving the back propagation algorithm	

Module 3: Support Vector Machines	**Lectures: 8**
The concept of optimal separation, Kernels and choosing the right kernel, The SVM algorithm, Extending SVM for multi class classification, SYM regression	

Module 4: Learning with Trees	**Lectures: 8**
Using decision trees, Constructing decision trees, Entropy, ID3, Dealing with continuous variables, Computational complexity, Gini Index and CART	

Module 5: Ensemble Learning	**Lectures: 4**
Concepts, Bagging, Boosting, Random Forest, Different ways to combine classifiers	

Textbook:

1 Marsland S., “Machine Learning: An Algorithmic Perspective”, CRC Press, 2nd Edition, 2015

Reference Books:

1) Han J., Kamber M., Pei J., “Data Mining: Concepts and Techniques”, Moran Kaufman, 3rd Edition, 2012

2) Zaki M. J., Meira W., “Data Mining and Analysis: Foundations and Algorithms”, 2nd Edition, Cambridge University Press, 2020

Mapping CourseOutcomesonto ProgramOutcomes

CO PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO12	PS O1	PSO2	PS O3
CO 1	High	Medium	Medium	Medium	Medium	High	Low	Low	Medium	Medium	No	Low	High	Medium	Medium
CO 2	High	High	High	High	High	High	Low	Low	Medium	Medium	No	Low	High	High	Medium
CO 3	High	High	High	High	High	High	Low	Low	Medium	Medium	No	Low	High	High	High
CO 4	High	High	High	High	High	High	Low	Low	Medium	Medium	No	Low	High	High	High
CO 5	High	High	High	High	High	High	Low	Low	Medium	Medium	No	Low	High	High	High

Course code: CS239
Course title: **Operating System**
Pre-requisite(s): **Data Structure, Computer System Architecture, Basic Course on Computer Programming**
Co- requisite(s):
Credits: **L:3 T:0 P:0**
Class schedule per week: **3**
Class: BTech
Branch: CSE/IT/AIML

Course Outcomes

After the completion of the course student will be able to:

1.	Describe the main components of OS and their working
2.	Explain the concepts of process and thread and their scheduling policies
3.	Solve synchronization and deadlock issues
4.	Compare the different techniques for managing memory, I/O, disk and files
5.	Design components of operating system

Module I: Overview and Processes **[8 Lectures]**
Operating system Objective and Functions, Evolution of Operating System, Major Advances in OS Components, Characteristics of Modern Operating Systems, Process Description and Control, Process Concept, Process States, Process Description, Process Control, Threads, Types of Threads, Multicore and Multithreading

Module II: Scheduling **[8 Lectures]**
Type of scheduling, Uniprocessor Scheduling, Multiprocessor Scheduling

Module III: Concurrency Mutual Exclusion and Synchronization **[8 Lectures]**
Principle of Concurrency, Mutual Exclusion, Hardware Support, Semaphores, Monitors, Message Passing, Readers/Writers Problem **Deadlock and Starvation,** Principle of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Dining Philosopher Problem

Module IV: Memory Management **[8 Lectures]**
Memory Management Requirements, Memory Partitioning, Paging, Segmentation, Virtual Memory Hardware and Control Structures, Operating System Policies for Virtual Memory

Module V: I/O Management, Disk Scheduling & Files **[8 Lectures]**
I/O device, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling, RAID, Disk Cache, File Management - Overview, File Organization and Access, File Directories, File Sharing, Record Blocking, File Allocation and Free Space Management

Text Book:

1. Stallings W., Operating systems - Internals and Design Principles, , 8th Edition, Pearson, 2014.

Reference Books:

1. Silberchatz Abraham, Galvin Peter B.,Gagne Greg, Operating System Principles, 9th Edition, Wiley Student Edition, 2013.
2. Tanenbaum Andrew S., Modern Operating Systems, 4th Edition, Pearson, 2014.
3. Dhamdhere D. M., Operating Systems A concept - based Approach, 3rd Edition, McGraw Hill Education, 2017.
4. Stuart B. L., Principles of Operating Systems, 1st Edition, 2008, Cengage learning, India Edition.
5. Godbole A. S., Operating Systems, 3rdEdition, McGraw Hill Education, 2017.

Course code: IT337
Course title: **Software Engineering**
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:0P:0 Class schedule per week: 3 Class: B. Tech
Semester / Level: VI / III
Branch: CSE/IT

Course Objectives
This course enables the students to:

1.	Students are effective team members, aware of cultural diversity, who conduct themselves ethically and professionally
2.	Students use effective communication skills and technical skills to assure production of quality software, on time and within budget.
3.	Students build upon and adapt knowledge of science, mathematics, and engineering to take on more expansive tasks.
4.	Able to increase level of self-reliance, technical expertise, and leadership.

Course Outcomes
After the completion of this course, students will be:

1.	Explain the software engineering principles and techniques
2.	Apply Software Project Management Practices
3.	Apply the knowledge gained for their project work as well as to develop software following software engineering standards
4.	Develop self-reliance, technical expertise, and leadership.

Module I
Introduction Some Definitions, FAQs about software engineering, the evolving role of software, Software process models, Waterfall model, the prototyping model, spiral model, RAD and Incremental model, Management activities, Project planning and Project Scheduling. (8L)

Module II
Software Requirements Functional and non-functional requirements, User requirements, System requirements, the software requirements document. IEEE standard of SRS, Quality of good SRS.

Requirement Engineering Process: Feasibility study, Requirements elicitation and analysis, Requirements validation, Requirement management. (8L)

Module III
Design Engineering
Design Process and Design Quality, Design Concepts, Design Models, Object oriented Design, UML: Class diagram, Sequence diagram, Collaboration diagram. (8L)
Module IV
Verification and Validation
Verification and Validation Planning, S/W inspection, static analysis.

Software Testing
Testing functions, Test care design, White Box testing, Black box testing, Unit testing, Integration Testing, System testing, Reliability. (8L)

Module V
Process metrics, Software Measurement, Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Quality assurance and standards, Quality planning, Quality control, S/W Maintenance in detail. (8L)

Text Book:
Sommerville, Software Engineering, 7th Edition, Pearson Education Publication. (T1)

Reference Books:

Pressman R. S.,Software Engineering: A Practiioners Approach, 5th Edition., TMA, New Delhi.(R1)

Mall Rajib, Fundamental of Software Engineering, 4th Edition, PHI Learning Private Limited.(R2)

Peters J. F. &Pedrycz W., Software Engineering, John Wiley & Sons, Inc. 2000.(R3)

Behforooz A. & Hudson F.J., Software Engineering Fundamentals, Oxford Univ. Press,New York, 2000.(R4)

Course code: **IT331**
Course title:**Image Processing**
Pre-requisite(s): **Discrete Mathematics, Data Structures**
Co- requisite(s):
Credits: **L: 3 T:0 P: 0**
Class schedule per week: **3**
Class: **B. Tech**
Branch: **CSE/IT/AIML**

Course Outcomes
After the completion of this course, students will be:

1.	Understand the concept of image formation, digitization, and role human visual system plays in perception of image data and spatial filtering techniques for enhancing the appearance of an image.
2.	Acquire an appreciation for various frequency-based filtering techniques for enhancing the appearance of an image, duly applying them in different applications.
3.	Discern the difference between noise models, gain an insight into assessing the degradation function and realize different spatial and frequency-based filtering techniques for reduction and removal of noise.
4.	Synthesize a solution to image compression using the concept of information theory and lossless and lossy compression techniques.
5.	Design and create practical solutions using morphological and image segmentation operators for common image processing problems and assess the results.

Module I **[8 Lectures]**
Introduction to Digital Image Processing, Elements of Visual Perception, Image Sensing & Acquisition, Sampling and Quantization, Basic Relationships between Pixels, Intensity Transformations, Histogram Processing, Spatial Convolution & Correlation, Smoothing Spatial Filters, Sharpening Spatial Filters.

Module II **[8 Lectures]**
Introduction to the Fourier Transform, Discrete Fourier Transform, Properties of the Two-Dimensional Fourier Transform, Image Smoothing using Frequency Domain filters, Image Sharpening using Frequency Domain filters, Selective Filtering, Basics of Fast Fourier Transform, Basics of: Walsh- Hadamard Transform; K-L Transform; Discrete Cosine Transform.

Module III **[8 Lctures]**
Model of Image Degradation/Restoration Process, Noise Probability Density Functions, Restoration in presence of Noise only, Periodic Noise Reduction using Frequency Domain filtering, Circulant Matrices, Block Circulant Matrices, Unconstrained Restoration, Constrained Restoration, Basics of Inverse Filtering

Module IV	**[8 Lectures]**
Image Compression Fundamentals – Coding Redundancy, Interpixel Redundancy, Psychovisual Redundancy, Fidelity Criteria, Image Compression Models– Source Encoder and Decoder, Channel Encoder and Decoder, Elements of Information Theory, Error-Free Compression – Variable-Length Coding, Bit-Plane Coding, Lossless Predictive Coding. Lossy Compression – Lossy Predictive Coding, Transform Coding. Color Fundamentals, Color Models, Basics of Full Colour Image Processing	

Module V	**[8 Lectures]**
Morphological Image Processing-Preliminaries, Dilation and Erosion, Opening and Closing, Hit-or-Miss Transformation, Boundary Extraction, Hole Filling, Connected Components, Convex Hull, Thinning, Thickening, Skeletons, Pruning Image Segmentation- Fundamentals,Point, Line and Edge Detection, Thresholding, Region Based Segmentation, Segmentation based on colour.	

Text books:

1. Rafael. C. Gonzalez & Richard E. Woods- Digital Image Processing, 3/e Pearson Education, New Delhi - 2009

Reference books:

1. W.K.Pratt-Digital Image Processing, 4/e, John Wiley & sons, Inc. 2006.
2. M. Sonka et al. Image Processing, Analysis and Machine Vision, 2/e, Thomson, Learning, India Edition, 2007.
3. Jayaraman, Digital Image Processing, Tata McGraw-Hill Education, 2011

Course code: AI309
Course title:Evolutionary Computing
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:1 P:0
Class schedule per week: 4
Class: B. Tech
Semester / Level: III
Branch: CSE/IT

Course Outcomes:
On completion of the course students shall be able to:

1. understand what kind of problems can be solved using EAs.
2. use the common EA operators and appreciate their need.
3. Solve real life problems using EAs
4. compare the performance of EAs with traditional algorithms.
5. perform multiobjective optimizations with constraints using EAs.

Module 1: Preliminaries	**Lectures: 6**
Module 1: Introduction to Evolutionary Computation Introduction, Problems that can be solved, Understanding the challenges of NP problems, Inspiration from biology, Advantages of evolutionary computing.	

Module 2: Basic Model (8L)	**Lectures: 6**
What is an evolutionary algorithm? Components of evolutionary algorithms, The operation of an evolutionary algorithm, Natural versus artificial evolution. Iterating through application problems.	

Module 3: Operators in Evolutionary Computations	**Lectures: 12**
Representation – Binary, Integer, Floating point, Permutation representations, Selection, Population Management, Mutation, Recombination, Elitism, Nicheism	

Module 4: Common Evolutionary algorithms	**Lectures: 8**
Problem solving with Hill climbing, Tabu Search, Simulated Annealing, Genetic Algorithms, Particle Swarm optimization.	

Module 5: Advanced Topics	**Lectures: 8**
Performance measures, Peak vs Average performance, Multiobjective problem solving, Constraint handling.	

Textbook:

1. Eiben A. E., Smith J. E., Introduction to Evolutionary Computing, 2nd Edition, Springer Publications, 2015.

Reference Books:

1. A. P. Engelbrecht, Computational Intelligence, Wiley, USA, 2007
2. Genetic Algorithms in Search, Optimization and machine Learning, D. E. Goldberg, Addition -Wisley Companay Inc, 1989
3. S.N. Sivanandam, S.N. Deepa, “Introduction to Genetic Algorithms”, Springer Berlin, Heidelberg, New York,2008

Course code: AI311
Course title: Network Analysis Lab.
Pre-requisite(s): MA 205, CS241
Co- requisite(s): Network Analysis Lab
Credits: L:3 T:0 P: 0
Class schedule per week: 3
Class: B. Tech
Semester / Level: Vth/III
Branch: AI/ML

Course Outcomes

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

1.	associate groups of data with type of network
2.	Ascertain mathematical properties of networks
3.	Implement simple visualisation and traversal algorithms for networks
4.	Categorize a network as a real world network.
5.	Decide whether a network is a random network.

Syllabus

Module 1: Introduction **[8 Lectures]**
Types of networks – Technological, Information, Social, Biological, Mathematics of networks – Representations, Weighted, Directed, Bipartite, Multilayer and Dynamic, Trees, Planar, Degree, Walks and Paths, Components, Independent paths, connectivity and cut sets, graph Laplacian

Module 2: Measures and Metrics **[8 Lectures]**
Centrality, Groups of nodes, Transitivity and clustering coefficient, Reciprocity, Signed edges and structural balance, Similarity, Homophily and assorative mixing.

Module 3: Algorithms for networks **[8 Lectures]**
Storing Network data, Algorithms for basic network quantities, Shortest path and breadth first search, Shortest paths in networks with varying edge lengths, Maximum flow and minimum cuts.

Module 4: Structure of Real world Networks **[8 Lectures]**
Components, components and small world effects, degree distribution, Power law and scale free networks, Distribution of other centrality measures, Clustering coefficients.

Module 5: Random Graphs **[8 Lectures]**
Random Graphs, Mean number of edges and mean degree, Degree distribution and clustering coefficient, Giant component, Small components, Path lengths

Course code: **AI 313**
Course title: **Classical Optimization Techniques**
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:0 P:0
Class schedule per week: 3
Class: B. Tech
Semester / Level: III
Branch: CSE/IT

Course Outcomes:

On completion of the course, students shall be able to:

CO1: Apply problem solving techniques through OR approaches.

CO2: Construct the operational models for the real-world applications using Linear Programming methods .

CO3: Analyse the optimal solution for the given problem by applying Integer Programming and Advanced LP solution Techniques.

CO4: Solve Assignment and Transportation Problems using classical techniques.

CO5: Model problems using Non-Linear Programming and evaluate the suitability of the available techniques for the problem at hand.

Module 1: Introduction to Operations Research	**Lectures: 6**
Operations Research (OR)An overview, Organ and Development of OR, Nature and Features of OR Modelling in OR, General Solution Methods for OR models, Scientific method in OR, Methodology of OR, Application, Opportunities and Shortcomings of OR.	
Module 2: Linear Programming	**Lectures: 10**
Introduction, Mathematical Formulation of the Problem, Graphical Solution Method, Some Exceptional Cases, General LPP, Canonical and Standard forms of LPP, Simplex Method: Introduction, Fundamental properties of solutions, the Computational Procedure, Use of Artificial variables, two-phase method, Big M method. Duality in linear programming, formulation of dual linear programming and examples.	
Module 3: Integer Programming and Advance LPP techniques	**Lectures: 8**
Introduction, Gomory's Method, Fractional Cut Method, Integer & Mixed Integer Problem , Branch and Bound Technique , Revised Simplex Method, Bounded Variable, Parametric LPP, Karmakar Algorithm.	
Module 4: Transportation and Assignment Problems:	**Lectures: 8**
Mathematical model of transportation problem, methods of finding initial solution (Northwest corner rule, Least cost method, Vogel's approximation method), test for optimality in TP using MODI Method. Mathematical model of Assignment problem, Hungarian method for solving Assignment problem.	
Module 5: Other Optimization modelling and Applications	**Lectures: 8**
Nonlinear Programming : Sample Applications , Graphical Illustrations of Nonlinear Programming, One-Variable and Multi-variable Unconstrained Optimization, The Karush-Kuhn-Tucker(KKT) Conditions. **Forecasting** : Forecasting Models, Judgemental Forecasting methods, Time Series Forecasting Methods.	

Textbook:
Hiller S. & Lieberman G.J. “Introduction to Operations Research”, 11th Edition, McGraw Hill, August 2021.

Reference Books:

1) Taha H.A. “Operations Research: An Introduction”, 10th Edition, Pearson August 2019.
2) Pai Pradeep Prabhakar “Operations Research”, 1st Edition, Oxford University Press 2012.

Course code: AI302
Course title: Supervised Learning Lab.
Pre-requisite(s):
Co- requisite(s): Supervised Learning
Credits: L: T: P: 3
Class schedule per week: 3
Class: B. Tech
Semester / Level: IVth/III
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	Able to preprocess data to improve classifier performance
2.	Able to implement basic classification algorithms and measure their efficacy.
3.	Able to implement simple tree based classifiers.
4.	Able to implement binary classifiers with a single hidden layer.
5.	Able to implement ensemble learning methods.

Module 1: Preprocessing **[8 Lectures]**
Implementation of algorithms for Normalizing data, Discretizing numeric features using thresholding and entropy, Measure correlation, Denoising data, Imputation of missing data, sample data sets.

Module 2: Basic Algorithms **[8 Lectures]**
Implementing Linear Regression using formula and iterative methods, Naïve Bayes classification, K-NN, Computing and visualizing performance metrics.

Module 3: Implementing Tree based classifiers **[8 Lectures]**
Computing entropy of features and datasets, Implementing basic tree classifiers,

Module 4: Implementing Neural Network **[8 Lectures]**
Computing loss function for binary classification, implementing back propagation for neural network using a single hidden layer. Performing multiclass classification using OvA.

Module 5: Ensemble Learning **[8 Lectures]**
Implementing a vote based classifier, Implementing a random forest. Estimating classifier improvement

COURSE INFORMATION SHEET

Course code: AI303
Course title: Unsupervised Learning
Pre-requisite(s):
Co- requisite(s): Unsupervised Learning Lab
Credits: L:3 T:0 P: 0
Class schedule per week: 3
Class: B. Tech
Semester / Level: VI/ III
Branch: AI/ML

Course Objectives
Thiscourseenablesthestudents to:

A.	Understand the different attribute types for correctly capturing the clusters.
B.	Correctly apply agglomerative clustering techniques.
C.	Apply and interpret results from partitional clustering methods.
D.	Understand the use of Density based clustering methods.
E.	Understand how to evaluate clustering results using a variety of metrics.

Course Outcomes

After the completion of this course, students will be:

1.	Able to demonstrate the basic concepts of clustering.
2.	Able to build trees to create clusters and define different distance measures.
3.	Able to perform iterative clustering.
4.	Able to decide when to perform density-based clustering.
5.	Able to estimate the estimate the performance of a clustering algorithm.

COURSE INFORMATION SHEET

Course code: AI303

Syllabus

Module 1: Introduction to Clustering **[8 Lectures]**
Introduction to Cluster Analysis, Capturing clusters, Need for visualizing data, Proximity matrix, Dendrograms

Module 2: Hierarchical Clustering **[8 Lectures]**
Single link vs Complete-link clustering, Agglomerative vs Divisive Clustering, Ward's Method, Probabilistic Hierarchical Clustering

Module 3: Partition based Clustering **[8 Lectures]**
Iterative Partition Cluster Method, The Initial Partition, K-Means Algorithm, K-Medoids Algorithm (PAM)

Module 4: Density based Clustering **[8 Lectures]**
Introduction, Algorithms for clustering categorical data – ROCK, Defining Density for clustering, DBSCAN.

Module 5: Estimating Cluster Validity **[8 Lectures]**
Internal, External and Relative validation. Purity, Maximum Matching, F-Measure, Entropy based measures. Pairwise measures including Jaccard Coefficient. Internal measures including BetaCV Measure, Dunn Index, Davies Bouldin index, Silhouette Coefficient. Relative measures including Calinski Harabasz index.

Text books:

1. King R., S., "Cluster Analysis and Data Mining: An Introduction", Mercury Learning and Information, 1st Edition, 2015.

Reference books:

2. Mohammed J. Zaki and Wagner Meira, Jr, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms" , 2nd Edition, Cambridge University Press, March 2020
3. Han J., Kamber M., "Data Mining Concepts and Techniques", 3rd Edition, Morgan Kauffman Press, 2019.

Mapping CourseOutcomesonto ProgramOutcomes

CO PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	High	Medium	Medium	Medium	Medium	High	Low	Medium	Medium	Medium	No	Medium	High	Medium	Medium
CO2	High	High	High	High	High	High	Low	Medium	Medium	Medium	No	Medium	High	High	High
CO3	High	High	High	High	High	High	Low	Medium	Medium	Medium	No	Medium	High	High	High
CO4	High	High	High	High	High	High	Low	Medium	Medium	Medium	No	Medium	High	High	High
CO5	High	High	High	High	High	High	Low	Medium	Medium	Medium	No	Medium	High	High	High

Course code: AI307 **Course title:** Modern Artificial Intelligence **Pre-requisite(s):** **Co- requisite(s):** **Credits:** L:3 T:0 P:0 **Class schedule per week**: 3 **Class:** B. Tech **Semester / Level**: VI/III **Branch**: CSE/IT/AIML

Course Objectives

This course enables the students:

1	To understand Artificial intelligence, task of AI, history of AI, concept of agent, understand different types of environment, application of AI
2	To understand the idea of search based problem solving, state space search, heuristic search, hill climbing, simulated annealing, search in complex environment
3	To understand how an agent can store and use knowledge to reason and make decision, basic principles of propositional logic, how to translate propositional into predicate logic, first order logic
4	To build intelligent system that can make decision in presence of uncertainty, understand Bayesian network, Probabilistic reasoning and Belief networks
5	To understand the fundamental concept of Machine learning, Neural networks and Deep learning

Course Outcomes

After the completion of this course, students will be able:

1	Determine problems that are amenable to AI-based solutions
2	Able to state and apply major algorithms, methods, and theoretical results in the field of artificial intelligence, Illustrate uninformed and informed search techniques for problem solving in intelligent systems.
3	Able to understand and use different knowledge representation techniques enabling reasoning and inference in artificial intelligence-based systems
4	Able to formulate , solve planning problems, planning types, reason in environment, understand Bayesian Inference, and Probabilistic reasoning,
5	Ability to implements machine learning algorithms, train and evaluate models, implement different activation functions and optimization techniques.

Module 1: Preliminaries	**Lectures: 6**
What is Artificial Intelligence (AI)? Evolution of AI, Intelligent Agents, Concept of rationality, Nature of environments, Structure of agents, Applications of AI	

Module 2: Automated Problem Solving	**Lectures: 8**
Search based problem solving, State space search, Heuristic search, Game tree search,	

Local search, Search in complex environments

Module 3: Knowledge Representation and Reasoning	**Lectures: 10**
Knowledge based agents, propositional logic, propositional logic to predicate logic, propositional logic-based agents, First order predicate logic, Knowledge representation in First Order Logic, Forward chaining, Backward chaining, Inference by resolution refutation	

Module 4: Planning and Probabilistic Reasoning	**Lectures: 8**
Planning in AI, Components of a planning problem, Types of planning in AI, Reasoning under uncertainty, Bayesian Inference, Probabilistic reasoning, Belief networks	

Module 5: Learning	**Lectures: 8**
Machine learning fundamentals, Forms of learning, learning decision trees, neural networks-based learning, Deep learning fundamentals	

Textbook:

1. Stuart Russell, Peter Norvig, Artificial intelligence: A Modern Approach, Prentice Hall, Fourth edition, 2020.

Reference Books:

1) Deepak Khemani, A First Course in Artificial Intelligence, Mc Graw Hill, First Edition, 2013.
2) Sridhar and Vijayalakshmi, Machine Learning, Oxford University Press, First Edition, 2021

Gaps in the syllabus (to meet Industry/Profession requirements):N/A
POs met through Gaps in the Syllabus:N/A
Topics beyond syllabus/Advanced topics/Design:N/A
POs met through Topics beyond syllabus/Advanced topics/Design:N/A

Course Outcome (CO) Attainment Assessment Tools and Evaluation Procedure Direct Assessment

Assessment Tools	**% Contribution during CO Assessment**
Continuous Internal Assessment	50
Semester End Examination	50

Continuous Internal Assessment	**% Distribution**
Mid semester examination	25
Two quizzes	20(2X10)
Teacher's Assessment	5

Assessment Components	CO1	CO2	CO3	CO4	CO5
Continuous Internal Assessment	√	√	√	√	√
Semester End Examination	√	√	√	√	√

Indirect Assessment
1. Student Feedback on Faculty
2. Student Feedback on Course

Course Delivery Methods

CD1	Lecture by use of boards/LCD projectors/OHP projectors
CD2	Assignments
CD3	Laboratory experiments/Teaching aids/Seminars
CD4	Mini projects
CD5	Industrial visits/in-plant training
CD6	Self- learning such as use of NPTEL materials and internets
CD7	Simulation

Mapping of Course outcomes onto Program Outcomes

Course Outcome	Program Outcomes (POs)												Program Specific Outcomes(PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
CO1	3	2	2	2	2	2	2	1	1	1	2	2	2	1	1
CO2	2	2	2	1	1	2	1	2	2	2	1	2	1	1	2
CO3	3	3	1	3	3	3	2	1	1	1	2	1	1	1	2
CO4	2	1	2	1	1	1	1	1	1	1	1	1	1	2	1
CO5	3	2	1	2	3	1	1	1	1	2	1	1	1	2	2

MAPPING BETWEEN COURSE OUTCOMES AND COURSE DELIVERY METHODS

Course Outcomes	Course Delivery Method
CO1	CD1, CD6
CO2	CD1, CD6, CD7
CO3	CD1, CD2, CD3, CD6, CD7
CO4	CD1, CD3, CD6, CD7
CO5	CD1, CD2, CD3, CD4, CD5, CD7

Course code: **AI 315**
Course Title: Advance Algorithm
Pre-requisite(s):
Co- requisite(s):
Credits: L:3 T:1 P:0
Class schedule per week: 3
Class: B. Tech
Semester / Level: VI
Branch: AI/ML

Course Outcomes:	
1)	Understand the concepts, properties, and analyse the operation of advanced Trees and can be able to solve real-world problems, such as database indexing and file systems.
2)	Able to apply these advanced algorithmic techniques to solve complex mathematical and computational problems.
3)	Able to understand advanced topics in combinatorial optimization and approximation algorithms.
4)	Able to solve efficiently diverse computational challenges.
5)	Able to analyse and design efficient and incentive-compatible solutions for various computational and strategic problems

Module 1: Advanced Data Structures and Analysis	**Lectures: 8**
Advanced Trees: B-trees, Red-Black Trees, Augmented Trees. Persistent Data Structures: Persistent Trees, Persistent Hashing. Van Emde Boas Trees: Overview and Operations. Amortized Analysis: Potential Method, Aggregate Analysis.	
Module 2: Advanced Divide and Conquer Algorithms	**Lectures: 8**
Strassen's Matrix Multiplication. Fast Fourier Transform: Cooley-Tukey Algorithm, Applications. Integer Multiplication: Karatsuba Algorithm, Toom-Cook Algorithm.	
Module 3: Advanced Greedy and Approximation Algorithms	**Lectures: 8**
Matroid Theory: Matroid Intersection, Greedy Algorithm with Matroids. Submodular Functions: Maximization and Minimization, Greedy Algorithms. PTAS and FPTAS: Polynomial-Time Approximation Schemes, Fully Polynomial-Time Approximation Schemes.	
Module 4: Advanced Dynamic Programming	**Lectures: 8**
Convex Hull Optimization: Line Container, Li-Chao Tree. Advanced Longest Common Subsequence: Space-Optimized, Rectangular LCS. All Pairs Shortest Paths: Floyd-Warshall Algorithm with Path Reconstruction. Range Queries and Updates: Segment Trees, Fenwick Trees, Lazy Propagation.	
Module 5: Advanced Randomized Algorithms & Game Theory	**Lectures: 8**
Randomized Algorithms for Linear Algebra: Randomized LU Decomposition, Randomized SVD. Randomized Online Algorithms: Ski Rental, Online Bin Packing. Randomized Parallel Algorithms: Randomized Parallel Sorting, PRAM Algorithms. **Game Theory**: Mechanism Design: Vickrey-Clarke-Groves Mechanisms, Incentive Compatibility.	

Algorithmic Aspects of Game Theory: Nash Equilibria, Price of Anarchy.

Textbook:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022.

Reference Books:

1. Donald E. Knuth, "The Art of Computer Programming, Volume 2: Seminumerical Algorithms", 3rd Edition, Pearson, 2005
2. Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms", Reprint Edition, Cambridge University Press, 2000.

SEMESTER VI LABORATORIES

Course code: AI304
Course title: Unsupervised Learning Lab.
Pre-requisite(s):
Co- requisite(s): Unsupervised Learning
Credits: L: T: P: 3
Class schedule per week: 3
Class: B. Tech
Semester / Level: IVth
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	Able to preprocess data to improve clustering performance
2.	Able to implement basic clustering algorithms and measure their efficacy.
3.	Able to implement simple tree based clustering.
4.	Able to implement density based clustering algorithms
5.	Able to implement advanced and non traditional clustering algorithms.

Module 1: Preprocessing **[8 Lectures]**
Experiments to denoise, standardize and perform other data wrangling tasks e.g. discretization. Compute distance between vectors using Euclidean, Chessboard, Manhattan and Minkowski's distance measures. Measure the entropy of a dataset. Vectorize distance computations

Module 2: Partitional Clustering **[8 Lectures]**
Perform Partitional clustering of a dataset using K-Means algorithms. Visualize the clusters in two and three dimensions. Calculate the entropy of the clustered dataset while varying k. Perform bisecting k-means. Implement medoid based clustering.

Module 3: Tree based classifiers **[8 Lectures]**
Implement an agglomerative clustering algorithm e.g. AGNES. Implement different measures to calculate the distance between clusters e.g. mean distance, nearest neighbour distance, farthest neighbour distance. Compare the results with partitional clusters for same number of clusters.

Module 4: Density based clustering **[8 Lectures]**
Implement density-based clustering using DBSCAN. Test the performance of the algorithm using non symmetric tor datasets. Measure the goodness of the clusters using standard methods like Silhouette coefficient, Ward's formula, Dunne's method etc.

Module 5: Advanced clustering **[8 Lectures]**
Cluster datasets using fuzzy clustering, GMM and evolutionary approaches like GA

Course code**: IT347**
Course title: **Introduction to Distributed System**
Pre-requisite(s):
Co- requisite(s):
Credits**: L: 3 T: 0 P: 0**
Class schedule per week: **3**
Class: B. Tech
Branch: CSE/IT/AIML

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

1.	Define distributed systems and their architecture.
2.	outline peer to peer services and distributed file systems
3.	Elaborate on concepts of process and resource management
4.	Analyze the requirements for designing and supporting distributed systems
5.	Discuss and design the working of distributed systems

Module I: Introduction **[8 Lectures]**
Introduction to "Large-Scale"distributed systems, Consequences of "large-scale" Some large-scale distributed systems, Architectures of large scale distributed systems.

Module II: Design principles of Distributed system **[8 Lectures]**
Introduction to peer-to-peer systems, The peer-to-peer paradigms, Services on structured overlays, Building trust in P2P systems.

Module III: Communication in Distributed system **[8 Lectures]**
System Model – Inter Process Communication – the API for internet protocols – External data representation and Multicast communication. Peer to Peer Services: Peer-to-peer Systems - Introduction - Napster and its legacy - Peer-to-peer - Middleware – Routing overlays. File System: Features-File model - File accessing models- File sharing semantics
Naming: Identifiers, Addresses, Name Resolution - Name Space Implementation - Name Caches - LDAP.

Module IV: Remote Method Invocation and Objects **[8 Lectures]**
Remote Invocation - Introduction - Request-reply protocols - Remote procedure call - Remote method invocation. **Case study:** Java RMI – Group communication - Publish-subscribe systems - Message queues - Shared memory approaches -Distributed objects - Case study: Enterprise Java Beans -from objects to components.

Module V: Process Management **[8 Lectures]**
Process Migration: Features, Mechanism - Threads: Models, Issues, Implementation.
Resource Management: Introduction- Features of Scheduling Algorithms -Task Assignment Approach - Load Balancing Approach - Load Sharing Approach.

Text Books:

1. Coulouris G., Dollimore J., and Kindberg T., "Distributed Systems Concepts and Design", 5th Edition, Pearson Education, 2012
2. Distributed Systems: Design and Algorithms, Editors(s):Serge Haddad, Fabrice Kordon, Laurent Pautet Laure Petrucci, Wiley online Library

Reference Books:

1. A. S. Tanenbaum, M. Van Steen, "Distributed Systems: Principles and Paradigms", Pearson Education, 2007
2. P. K. Sinha, "Distributed Systems: Concepts and Design", Prentice Hall, 2007.

Course code: **IT445**
Course title: **Internet of Things(IoT)**
Pre-requisite(s):
Co-requisite(s):
Credits: **L:3 T:0 P:0**
Class schedule per week: **3**
Class: B. Tech
Branch: CSE/IT/AIML

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

1.	Identify the IoT Components and its capabilities
2.	Explain the architectural view of IoT under real world constraints
3.	Analyse the different Network and link layer protocols
4.	Evaluate and choose among the transport layer protocols
5.	Design an IoT application

Module I; Introduction to IOT [8 Lectures]
The definition of the Internet of Things, main assumptions and perspectives. Platform for IoT devices Device architectures. Conventional and renewable power sources for resource-constrained devices. Operating systems for resource-constrained devices.

Module II: Architecture of IOT [8 Lectures]
Node structure: Sensing, Processing, Communication, Powering IOT networking: Topologies, Layer/Stack architecture, The data link layer for IoT- Wireless communication technologies. Wire communication technologies. Manet Networks.

Module III: Communication Technologies [8 Lectures]
Introduction to ZigBee, BLE, WiFi, LTE, IEEE 802.11ah, Discuss data rate, range, power, computations/bandwidth, QoS, Service oriented protocols (COAP). Communication protocols based on the exchange of messages (MQTT). Service discovery protocols.

Module IV: M2M and IoT Technology Fundamentals [8 Lectures]
Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management. (8L)

Module V: The data processing for IoT [8 Lectures]
Organization of data processing for the Internet of things. Cloud computing. Fog computing. Application case studies: Smart Grid. Home Automation. Smart City.

Text books:

1. Madisetti Vijay and BahgaArshdeep, Internet of Things (A Hands-on-Approach), 1st Edition, VPT, 2014.

2. Raj Pethuru and Raman Anupama C., The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press.

Reference books:

1. Vermesan Dr. Ovidiu, Friess Dr. Peter, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers.
2. Holler Jan, Tsiatsis Vlasios, Mulligan Catherine, Avesand Stefan, Karnouskos Stamatis, Boyle David, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st Edition, Academic Press, 2014.

Course code: **IT351**
Course title: **Natural Language Processing**
Pre-requisite(s):
Co- requisite(s):
Credits: **L: 3 T: 0 P: 0**
Class schedule per week: **3**
Class: **B. Tech**
Branch: **CSE/IT/AIML**

Course Outcomes

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

1.	Describe the typical NLP problem, their importance & difficulty; and concepts of morphology, syntax, semantics, discourse & pragmatics of natural language.
2.	Demonstrate understanding of the relationship between NLP and statistics& machine learning.
3.	Discover various linguistic and statistical features relevant to the basic NLP task, namely, spelling correction, morphological analysis, parts-of-speech tagging, parsing and semantic analysis.
4.	Analyse NLP problems to decompose them into appropriate components.
5.	Evaluate a NLP system, identify shortcomings and suggest solutions for these shortcomings.

Module I: Introduction to NLP **[8 Lectures]**
Introduction and applications, NLP phases, Difficulty of NLP including ambiguity; Spelling error and Noisy Channel Model; Concepts of Parts-of-speech and Formal Grammar of English.

Module II: Language Modelling **[8 Lectures]**
N-gram and Neural Language Models, Language Modelling with N-gram, Simple N-gram models, Smoothing(basic techniques), Evaluating language models;Neural Network basics, Training; Neural Language Model, Case study: application of neural language model in NLP system development**.**

Module III: POS Tagging **[8 Lectures]**
Parts-of-speech Tagging: basic concepts; Tagset; Early approaches: Rule based and TBL; POS tagging using HMM, POS Tagging using Maximum Entropy Model.

Module IV: Parsing Basic concepts **[8 Lectures]**
Top down and bottom up parsing, Treebank; Syntactic parsing: CKY parsing; Statistical parsing basics: Probabilistic Context Free Grammar (PCFG); Probabilistic CKY Parsing of PCFGs.

Module V: Semantics **[8 Lectures]**
Vector Semantics; Words and Vector; Measuring Similarity; Semantics with dense vectors; SVD and Latent Semantic Analysis; Embeddings from prediction: Skip-gram and CBOW; Concept of Word Sense;

Introduction to WorldNet.

Text books:

1. Jurafsky Dan and Martin James H., Speech and Language Processing **(3rd ed.)**Available at: https://web.stanford.edu/~jurafsky/slp3/.

Reference books:

1. Jurafsky D. and Martin J. H., Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd Edition, Upper Saddle River, NJ: Prentice-Hall, 2008
2. Goldberg Yoav, A Primer on Neural Network Models for Natural Language Processing.

Course code: **IT349**
Course title: **Cryptography and Network Security**
Pre-requisite(s):
Co- requisite(s):
Credits: **L:3 T:0 P:0**
Class schedule per week: **3**
Class: **B. Tech**
Branch: **CSE/IT/AIML**

Course Outcomes

After the completion of this course, students will be:

1.	Understand the basic concept of Cryptography and Network Security and their mathematical models, and to be familiar with different types of threats
2.	Learning and applying various Ciphering Techniques.
3.	Apply Symmetric and Asymmetric Cryptographic Algorithms and Standards in Networks.
4.	Examine the issues and structure of Authentication Service and Electronic Mail Security
5.	To explain and classify different malicious programs, worms and viruses, and to learn the working and design principles of Firewalls

Module I: Introduction to Cryptography [8 Lectures]

Computer Security concepts, The OSI Security Architecture, Security Attacks, Security Services, A model for Network Security, Classical Encryption Techniques.

Module II: Mathematical Foundations of Cryptography [8 Lectures]

Modular Arithmetic, Euclidean Algorithm, Groups, Rings, Fields, Finite Fields of the Form GF(p), Polynomial Arithmetic, Finite Fields of the Form GF(2^n), Prime Numbers, Fermat's and Euler's Theorem, The Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithms.

Module III: Symmetric and Asymmetric Cryptography [8 Lectures]

Difference Between Symmetric and Asymmetric Cryptography, DES, Triple DES, AES, RSA Cryptosystem, Symmetric and Asymmetric Key Cryptography Together, Elgamal Cryptosystem, Elliptic Curve Cryptosystems, , Diffie-Hellman Key Exchange , Cryptographic Hash Functions, Message Authentication Codes, Digital Signature.

Module IV: Internet Security Protocols [8 Lectures]

Basic Concepts, Security Socket Layer (SSL), Secure Hyper Text Transfer Protocol (SHTTP), Time stamping Protocol(TSP), Secure Electronic Transaction(SET), SSL Versus SET, 3-D Secure Protocol, Electronic Money, Email Security, Wireless Application Protocol(WAP) Security, Security in GSM.

Module V: Network Security [8 Lectures]

Users, Trusts and Trusted Systems, Buffer Overflow and Malicious Software, Malicious Programs, Worms, Viruses, Intrusion Detection Systems (IDS), Firewalls: Definitions, Constructions and Working Principles.

Text Book:

1. Forouzan B. A., Mukhopadhyay D., “Cryptography and Network Security”, 3^{rd} Edition, Mcgraw Higher Education, 2016.

Reference Books:

1. Stallings W., “Cryptography and Network Security: Principles and Practice”, 7^{th} Edition, Pearson, 2017
2. Kahate A., “Cryptography and Network Security”, 3^{rd} Edition, McGraw Hill Education, New Delhi, 2013
3. Schneier B., “Applied Cryptography: Protocols, Algorithms And Source Code In C”, 2^{nd} Edition, Wiley, 2007

Course code: AI401
Course title: Reinforcement Learning
Pre-requisite(s):
Co- requisite(s): Machine Learning Basics
Credits: L: 2 T: P:
Class schedule per week: 3
Class: B. Tech
Semester / Level: VII/ IV
Branch: AI/ML

Course Outcomes

After the completion of this course, students will be:

1.	Able to Explain basic mathematical concepts of Reinforcement learning
2.	Able to Solve real world problems using Bellman Optimality equations.
3.	Able to implement Monte Carlo Methos
4.	Able to implement TD learning
5.	Able to Solve problems related to Policy Gradient methods.

Module 1: Basic Concepts **[8 Lectures]**

A grid world example , State and action, State transition , Policy, Reward, Trajectories, returns, and episodes,Markov decision processes, Summary
State Values and Bellman Equation ,Motivating example1:Why are returns important? .Motivating example2:How to calculate returns? State values , Bellman equation Examples for illustrating the Bellman equation, Matrix-vector form of the Bellman equation, Solving state values from the Bellman equation ,From state value to action value , Summary.

Module 2: OptimalStateValuesandBellmanOptimalityEquation **[8 Lectures]**

Motivating example: How to improve policies? Optimal state values and optimal policies, Bellman optimality equation, Solving an optimal policy from the BOE, Factors that influence optimal policies, Summary
Value iteration, Policy iteration ,Truncated policy iteration, Summary

Module 3: Monte Carlo Methods: **[8 Lectures]**

Motivating example: Mean estimation, MC Basic, MC Exploring Starts, MC-Greedy:Learning without exploring starts, Exploration and exploitation of-greedy policies ,Summary
Stochastic Approximation: Motivatinge Example: Mean estimation, Robbins-Monro algorithm , Dvoretzkys convergence theorem , Stochastic gradient descent, Summary .

Module 4: Temporal Difference Learning **[8 Lectures]**

TD learning of state values, TD learning of action values: Sarsa, TD learning of action values: n-step Sarsa, TD learning of optimal action values: Q-learning, A unified viewpoint, Summary .

Value Function Methods :Value representation: From table to function,TD learning of state values based on function approximation, TD learning of action values based on function approximation, Deep Q-learning, Summary.

Module 5: Policy Gradient Methods **[8 Lectures]**

Policy representation: From table to function, Metrics for de fining optimal policies, Gradients of the metrics, Monte Carlo policy gradient(REINFORCE) . Actor-Critic Methods: The simplest actor-critic algorithm(QAC),Advantage actor-critic(A2C) Off-policy actor-critic, Deterministic actor-critic, Summary