UNIVERSITY SCHOOL

OF

INFORMATION AND COMMUNICATION TECHNOLOGY

Department of Information Technology

COURSE STRUCTURE

Course: Master of Computer Applications
(Specialization in Artificial Intelligence)

BATCH: 2025-27

GAUTAM BUDDHA UNIVERSITY
GAUTAM BUDH NAGAR, GREATER NOIDA, UP, INDIA

BRIDGE COURSE

S No.	Course Code	Course Name	Teaching Scheme			Credits	Types
			L	T	P		
1.	MCB001	Computer Fundamental and Programming	3	1	0	4	BC
2.	MCB003	Introduction to Internet Technology	3	0	0	3	BC
3.	MCB005	Fundamental of Operating Systems	3	0	0	3	BC
4	MCB081	Operating System Lab	0	0	4	2	BC-L
5	MCB083	Internet Technology Lab	0	0	4	2	BC-L
Total Hours and Credits			**9**	**1**	**8**	**14**	

*BC: Bridge Course, BC-L: Bridge Course Lab

SEMESTER I

S. No.	Course Code	Course Name	Teaching Scheme			Credits	Types
			L	T	P		
1.	MAI101	Computer Fundamental and C Programming	3	0	0	3	CC
2.	MAI103	Software Engineering	3	0	0	3	CC
3.	MAI105	Artificial Intelligence	3	0	0	3	CC
4.	MAI107	Discrete Mathematics	3	0	0	3	CC
5.	MAI109	Python Programming	3	0	0	3	CC
6.	MAI181	C Programming Lab	0	0	2	1	CC-L
7.	MAI183	Python Programming Lab	0	0	2	1	CC-L
8.	MAI185	Artificial Intelligence Using Prolog Lab	0	0	2	1	CC-L
9.	GP	General Proficiency	Non Credit				
Total Hours and Credits			**15**	**0**	**6**	**18**	

*CC: Core Course, CC-L: Core Courses Lab

SEMESTER II

S. No	Course Code	Course Name	Teaching Scheme			Credits	Types
			L	T	P		
1.	MAI102	Analysis and Design of Algorithm	3	0	0	3	CC
2.	MAI104	Data Structures	3	0	0	3	
3.	MAI06	Machine Learning	3	0	0	3	CC
4.	MAI108	Database Management System	3	0	0	3	CC
5.	MAI110	Natural Language Processing	3	0	0	3	CC
6.	MAI112	Theory of Computation	3	0	0	3	CC
7.	MAI184	Machine Learning Lab	0	0	2	1	CC-L
8.	MAI186	Database Management System Lab	0	0	2	1	CC-L
9.	MAI188	Natural Language Processing Lab	0	0	2	1	CC-L
10.	ITV301	Professional Ethics	2	0	0	1	VAC
11.	GP	General Proficiency	Non Credit				
Total Hours and Credits			**20**	**0**	**6**	**22**	

*VAC: Values added courses

SEMESTER I

Computer Fundamental and C Programming			
Course Code:	MAI101	Course Credits:	3
Course Category:	CC	Course (U / P)	P
Course Year (U / P):	1P	Course Semester (U / P):	1P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L + T):	45	End Sem. Exam Hours:	3

COURSE OBJECTIVES
1. To provide knowledge of primary and derived data types used in C
2. To make them understand basic conditional and break statements used in C
3. To provide a basic understanding of pointers and pointers arithmetic
4. To enable the students to explore how pre-defined functions are used and also created in a program
5. Learn the difference between static and dynamic memory allocation methods and also learn various dynamic memory allocation methods.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand the basic building blocks of C language like tokens, identifiers, constants and variables.
2. Acquire knowledge of various conditional and loop statements
3. Judge which data structure to use among arrays, struct and union depending on the application
4. Use pointers and tell the difference between call by value and call by reference.
5. Use dynamic memory allocation to create arrays, structures and union and also perform various operations on them.

UNIT I INTRODUCTION TO COMPUTER AND PROGRAMMING CONCEPTS

Definition, characteristic, generation of computers, basic components of a computer system, memory, input, output and storage units, high level language and low-level language, Soft- ware: system software, application software, hardware. firmware, Operating System, compiler, interpreter and assembler, linker, loader, debugger, IDE. Introduction to algorithm and flowchart; representation of algorithm using flowchart symbol, pseudo code, basic algorithm de- sign, characteristics of good algorithm, development of algorithm.

UNIT II INTRODUCTION TO C PROGRAMMING LANGUAGE

Introduction to C programming language, declaring variables, preprocessor statements, arithmetic operators, programming style, keyboard input, relational operators, introduction, feature of C language, concepts, uses, basic program structure, simple data types, variables, constants, operators, comments, control flow statement: if, while, for, do-while, switch.

UNIT III DATA TYPES AND STRUCTURES

Bitwise operators, Predefined and User defined data types, arrays, declaration and operations on arrays, searching and sorting on arrays, types of sorting, 2D arrays, passing 2D arrays to functions, structure, member accessing, structure and union, array of structures, functions, declaration and use of functions, parameter passing, recursion.

UNIT IV FUNDAMENTALS OF POINTERS

Introduction to pointers, pointer notations in C, Declaration and usages of pointers, operations that can be performed on computers, use of pointers in programming exercises, parameter passing in pointers, call by value, call by references, array and characters using pointers, dynamic memory allocation

UNIT V FILE HANDLING IN C AND ENUM

Introduction to file handling, file operations in C, defining and opening in file, reading a file, closing a file, input output operations on file, counting: characters, tabs, spaces, file opening modes, error handling in input/output operations, Enumerated data types, use of Enum, declaration of Enum.

Text Books:

1. **Herbert Schildt**, *C: The Complete Reference*, McGraw Hill Education, Latest Edition.
2. **Brian W. Kernighan and Dennis M. Ritchie**, *The C Programming Language* (2nd Edition), Pearson Education, 1988.
3. **E. Balagurusamy**, *Programming in ANSI C*, Tata McGraw Hill Education, Latest Edition.
4. **Alan R. Feuer**, *The C Puzzle Book: Puzzles for the C Programming Language*, Prentice Hall, 1982.
5. **Peter Van Der Linden**, *Expert C Programming: Deep C Secrets*, Dorling Kindersley (India), Latest Edition.

SOFTWARE ENGINEERING			
Course Code:	**MAI103**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester(U/ P)**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Knowledge of basic SW engineering methods and practices and application.			
2. A general understanding of software process models.			
3. Understanding of software requirements and the SRS documents.			
4. Understanding of software design process.			
5. Understanding of software coding, testing and maintenance.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Basic knowledge and understanding of the analysis and design of complex systems.			
2. Ability to apply software engineering principles and techniques.			
3. Ability to develop, maintain and evaluate large-scale software systems.			
4. To produce efficient, reliable, robust and cost-effective software solutions.			
5. Ability to perform independent research and analysis.			

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering: definitions, role of software engineering, planning a software project, defining the problem, developing a solution strategy, planning the development process, software engineering process paradigms, principles of software engineering, software engineering activities.

UNIT II SOFTWARE LIFE CYCLE MODELS

Software Development Life Cycle (SDLC), SDLC models, waterfall model and its variations, prototype model, iterative enhancement model, spiral model, RAD model, comparison of these models, software development teams, software development environments, validation and traceability, maintenance, prototyping requirements, Software project management.

UNIT III REQUIREMENT ANALYSIS AND DESIGN

Software Requirement Specification (SRS): Introduction, need of SRS, significance, characteristics of SRS, Structure of SRS, IEEE standards for SRS design, functional and non-functional requirements, Requirement gathering and analysis, requirement engineering and management.

UNIT IV SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design, bottom-up design, design methods: data-flow model: data flow diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, object models: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling.

UNIT V SOFTWARE CODING, TESTING AND MAINTENANCE

Coding, Testing Methods: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing, thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, Evolution of software products, economics of maintenance, category of software maintenance, Role of product development life cycle, deployment model, adaptive maintenance, corrective maintenance, perfective maintenance, enhancement request, proactive defect prevention, problem reporting, problem resolution, software maintenance from customers' perspective, maintenance standard: IEEE-1219, ISO-12207.

Text Books:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.
3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.

Reference books

1. Roger S. Pressman. Software Engineering: A Practitioner's Approach. McGraw-Hill Inc., 2004.
2. Nasib S. Gill. Software Engineering: Software Reliability, Testing and Quality Assurance Khanna Book Publishing Co (P) Ltd., New Delhi, 2002

ARTIFICIAL INTELLIGENCE			
Course Code:	**MAI105**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/ P)**	**1P**
No. of Lectures+ Tutorials(Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L+T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES
1. Gain a historical perspective of AI and its foundations.
2. Become familiar with basic principles of AI toward problem-solving, inference, perception, knowledge representation, and learning.
3. Investigate applications of AI techniques in intelligent agents, expert systems, and machine learning models
4. .Experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool.
5. Explore the current scope, potential, limitations, and implications of intelligent systems.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents
2. Analyze and formalize the problem as a state space, graph, design heuristics, and select different search or game-based techniques to solve them.
3. Develop intelligent algorithms for constraint satisfaction problems and also design intelligent .systems for Game Playing
4. Attain the capability to represent various real-life problem domains using logic-based techniques and use this to perform inference or planning.
5. Solve reasoning problems with Expert Systems.

UNIT I INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Basic concept of artificial intelligence (AI), history of AI, AI and consciousness, weak and strong AI, physical symbol system hypothesis, comparison of computer and human skills, practical systems based on AI, development of logic, components of AI, Turing Test in AI, Advantages and Disadvantages of AI, Intelligence, Intelligent System, Role of IS, Comparison of various IS, Mind-Body Problem in AI, Chinese Room Experiment in AI, Parallel and Distributed AI.

UNIT II PROBLEM SOLVING THROUGH AI

Defining the problem as state-space search, analyzing the problem, representing the problems from AI viewpoint, production system, developing production rules, characteristics of the production system, algorithm for problem-solving using AI technique.

UNIT III SEARCH TECHNIQUES

Use of search in AI problem solution, blind search techniques, heuristic search techniques, concept of heuristic knowledge, designing of the heuristic function, types of heuristic search techniques: generate and test, best first search, problem reduction using AND-OR graph, local search technique, branch and bound search, memory bounded search technique, local beam search, properties of heuristic search techniques, overestimation and underestimation of heuristic function hill climbing search, simulated annealing search, constraint satisfaction means ends analysis, Tic-Tac Toe Problem, Water Jug problem, Chess Problem, Tower of Hanoi problem, Travelling Salesman problem, Monkey and Banana Problem, Magic Square.

UNIT IV INTRODUCTION TO LOGIC

Introduction, proposition calculus, syntax of propositional calculus, semantics of propositional calculus, well-formed formula, properties of statements, inferencing of propositional logic, predicate logic, syntax of predicate logic, semantics of predicate logic, concept of resolution, resolution algorithm, skolemization, types of resolution unit resolution, binary resolution.

UNIT V AI TECHNIQUES AND APPLICATIONS

Introduction to Machine Learning, Introduction to Deep Learning, Introduction to Expert system: Introduction phases in building expert systems, Expert system versus traditional systems, rule-based expert systems, blackboard systems, application of expert systems, list of shells and tools, Introduction to Natural Language Processing, AI in future, AI in social Media, AI in Entertainment and education, AI in drones, AI in Automated Computer support, AI in personalized shopping experience, AI in Finance, AI in smart Cars, AI in travel and navigation, AI in smart home devices, AI in security and surveillance, AI in education, AI in health care, AI in E commerce.

Text Books:

1. Artificial Intelligence, Elanie Reich: Tata mcgraw Hill publishing house, 2008.
2. Artificial Intelligence, Peterson, TataMcGraw Hill, 2008.

Reference books:

1. Artificial Intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
2. Artificial Intelligence, Winston, PHI publication, 2006

DISCRETE MATHEMATICS			
Course Code:	**MAI107**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P)**	**1P**
No. of Lectures +Tutorials(Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES
1. Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.
2. Express a logic sentence in terms of predicates, quantifiers, and logical connectives.
3. Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.
4. Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.
5. Apply rules of inference, tests for validity, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. To express a logic sentence in terms of predicates, quantifiers, and logical connectives.
2. Apply the rules of inference, proof by contradiction, and mathematical induction.
3. Students will be able to evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
4. Students will be able to learn about predicates, quantifiers, and logical connectives.
5. Students will be able to use tree and graph algorithms to solve problems.

UNIT I MATHEMATICAL LOGIC

Statements and notations, connectives, well formed formulas, truth tables, tautology, equivalence implication, normal forms, predicates: predicative logic, free & bound variables, rules of inference, consistency, proof of contradiction, automatic theorem proving.

UNIT II SET THEORY

Set Theory: Introduction, Combination of sets, Multi sets, ordered pairs, Set Identities, Properties of binary relations, equivalence, compatibility and partial ordering relations, Hasse diagram. functions: Operations on functions, inverse function Classification of functions, recursive functions, lattice and its properties. algebraic structures: algebraic systems examples and general properties, semi groups and monads, groups sub groups" homomorphism, isomorphism.

UNIT III ELEMENTARY COMBINATORICS

Basis of counting, combinations & permutations, with repetitions, constrained repetitions, binomial coefficients, binomial multinomial theorems, the principles of inclusion – exclusion, pigeon hole principles and it's application.

UNIT IV RECURRENCE RELATION

Generating functions, function of sequences calculating coefficient of generating function, recurrence relations, solving recurrence relation by substitution and generating funds, characteristics roots solution of in homogeneous recurrence relation.

UNIT V GRAPH THEORY

Representation of graph, Trees: Definition, Binary tree, Binary tree traversal, Binary search tree. DFS, BFS, spanning trees, planar graphs. graph theory and applications, basic concepts isomorphism and sub graphs, multi graphs and euler circuits, hamiltonian graphs, chromatic numbers.

Text Books:

1. Discrete and Combinational Mathematics- An Applied Introduction-5th Edition – Ralph. P.Grimaldi, Pearson Education
2. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar.P, TMH
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.

Reference book:

1. Discrete Mathematical structures Theory and application-Malik & Sen
2. Discrete Mathematics for Computer science, Garry Haggard and others, Thomson.
3. Logic and Discrete Mathematics, Grass Man & Trembley, Person Education

PYTHON PROGRAMMING			
Course Code:	**MAI109**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/ P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L+ T):	**45+00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Master the fundamentals of writing Python scripts.			
2. Learn core Python scripting elements such as variables and flow control structures.			
3. Discover how to work with lists and sequence data.			
4. Write Python functions to facilitate code reuse.			
5. Use Python to read and write files.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Problem solving and programming capability.			
2. Explain basic principles of Python programming language			
3. Implement database and GUI applications.			
4. Implement object oriented concepts			
5. Define and demonstrate the use of built-in data structures "lists" and "dictionary"			

UNIT I PYTHON BASICS, CONDITIONAL &LOOPS

Installation of Python and python Notebook, Python Objects, Number & Booleans, Strings, Container objects, Mutability of objects, Operators - Arithmetic, Bitwise, comparison and Assignment operators, Operators Precedence and associativity. Conditions (If else, if-elif-else), Loops (While ,for), Break and Continue statements, Range Functions

UNIT II STRING OBJECTS AND LIST OBJECTS

String object basics, String methods, Splitting and Joining Strings, String format functions, list object basics, list methods, List as stack and Queues, List comprehensions,

UNIT III TUPLES, SET, DICTIONARIES & FUNCTIONS

Tuples, Sets, Dictionary Object basics, Dictionary Object methods, Dictionary View Objects. Functions basics, Parameter passing, Iterators, Generator functions, Lambda functions, Map, Reduce, filter functions

UNIT IV OOPS CONCEPTS & WORKING WITH FILES

OOPS basic concepts, creating classes and Objects, Inheritance, Multiple Inheritance, working with files, Reading and writing files, Buffered read and write, Other File methods

UNIT V MODULES, EXCEPTION HANDLING & DATABASE PROGRAMMING

Using Standard Module, Creating new modules, Exceptions Handling with Try-except, Creating, inserting and retrieving Table, Updating and deleting the data. Data Ananlysis- Numpy variable, Numpy manipulation, Scipy, Pandas intro. Descriptive analysis, Pandas Input-output, Pandas manipulation, Pandasgroupby

Text Books:

1. Head First Python 2e: A Brain-Friendly Guide Paperback – Illustrated, 16 by Paul Barry, Oreilly
2. Python: The Complete Reference Paperback – 20 March 2018 by Martin C. Brown (Author), TMH Publication

Reference book:

1. Let Us Python by Yashavant Kanetkar , 1 January 2019, BPB publication
2. Python Programming, A modular approach , First Edition, By Pearson Publication by Taneja Sheetal and Kumar Naveen , 26

DATA STRUCTURE LAB			
Course Code:	**MAI181**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Introduce the concept of data structures through ADT including List, Stack, Queues .			
2. To design and implement various data structure algorithms.			
3.To introduce various techniques for representation of the data in the real world.			
1. To develop application using data structure algorithms			
1. Compute the complexity of various algorithms.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Select appropriate data structures as applied to specified problem definition			
2. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.			
3. Students will be able to implement Linear and Non-Linear data structures.			
4. Implement appropriate sorting/searching technique for given problem.			
5. Design advance data structure using Non-Linear data structure			

List of Experiments:

1. Run time analysis of Fibonacci Series
2. Study and Application of various data Structure
3. Study and Implementation of Array Based Program
    a. Searching (Linear Search, Binary Search)
    b. Sorting (Bubble, Insertion, Selection, Quick, Merge etc)
    c. Merging
4. Implementation of Link List
    a. Creation of Singly link list, Doubly Linked list
    b. Concatenation of Link list
    c. Insertion and Deletion of node in link list
    d. Splitting the link list into two link list
5. Implementation of STACK and QUEUE with the help of
    a. Array
    b. Link List
6. Implementation of Binary Tree, Binary Search Tree, Height Balance Tree
7. Write a program to simulate various traversing Technique
8. Representation and Implementation of Graph
    a. Depth First Search
    b. Breadth First Search
    c. Prims Algorithm
    d. Kruskal's Algorithms
    e. Implementation of Hash Table

PYTHON PROGRAMMING LAB			
Course Code:	**MAI183**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U /P):**	**1P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 hours**

COURSE OBJECTIVES
1. To introduce students to the basic concepts and techniques of Machine Learning
2. A general understanding of ML process models
3. To introduce students to the basic concepts and techniques of Machine Learning
4. Understanding of Python Programming and its module
5. Understanding of Deep Learning.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Installation of python and its module &ipython notebook.
2. Ability to apply Python principles and techniques.
3. Ability to design ML models and test and train data set .
4. To Understand working of tensorflow.
5. Ability to perform deep learning algorithms.

LIST OF PRACTICALS

1. To print the largest/smallest of two numbers
2. To input three numbers and print the greatest of all
3. To read two numbers x and n and print xn (first write with the use of operator and then write with the help of inbuilt function
4. To input the value of x and n and print the sum of the series: 1+x+x2+x3+x4+…….xn
5. To check if a number is a perfect number or not
6. Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
7. Write a program to count the numbers of characters in the string and store them in a dictionary data structure
8. To print factorial of a number using function
9. To print factorial of a number using recursion
10. To count no of vowels in a string that was given as input by user
11. Write a function to find all duplicates in the list.
12. Write a function unique to find all the unique elements of a list.
13. Write a program to perform addition of two square matrices
14. Write a program to perform multiplication of two square matrices
15. To read from a text file and print each word separated by # symbol, example #vipin #

ARTIFICIAL INTELLIGENCE USING PROLOG LAB			
Course Code:	**MA1185**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester**	**1P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 hours**

COURSE OBJECTIVES
1. Become familiar with the basic knowledge of prolog.
2. Basic understanding of AI based Problems.
3. Basic idea of how to program in prolog and its working environment.
4. Understanding how to write a program using the rules.
5. Understanding the basic knowledge of how to create a expert system using prolog
COURSE OUTCOMES
At the end of the course the students should be able to:
1. be able to write programs in Prolog using techniques such as accumulators and difference structures
2. know how to model the backtracking behavior of program execution.
3. Appreciate the unique perspective Prolog gives to problem solving and algorithm design
4. Understand how larger programs can be created using the basic programming techniques used in this course
5. Understanding the basic knowledge of designing of Expert system.

EXPERIMENTS:

1. Write a program that verifies whether a line segment is horizontal, vertical or oblique.
2. Write a program to perform Map Colouring.
3. Write a program for Two Factorial Definitions.
4. Write a program to Loading & Editing a Program.
5. Write a program to sort a List using PROLOG.
6. Write a program for Depth first Search using PROLOG
7. Write a program for Simple Input-Output.
8. Write a program to copy the input of characters.
9. Write a program for Towers of Hanoi Puzzle.
10. Write a program for the Monkey and Banana Problem.

SEMESTER II

ANALYSIS AND DESIGN OF ALGORITHM			
Course Code:	**MAI102**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45+00**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES
1. To introduce fundamental concepts of algorithm analysis:
2. To familiarize students with major algorithmic design paradigms:
3. To demonstrate the application of algorithms to solve real-world problems:
4. To develop the ability to design and analyze efficient algorithms:
5. This objective aims to provide an understanding of the limitations of algorithms and the challenges of solving computationally difficult problems.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Ability to analyze algorithm performance:
2. Competence in solving common algorithmic problems:
3. Understanding of computational complexity:
4. Proficiency in applying algorithmic design techniques:
5. Skill in evaluating and comparing algorithms:

UNIT I BASIC CONCEPT OF ALGORITHMS

What is an algorithm, notion of algorithm, fundamentals of algorithmic solving, Mathematics for Algorithmic sets, Functions and Relations, Vectors and Matrices, linear Inequalities and Linear Equations, fundamentals of analysis framework, the efficient algorithm, Average, Best and Worst case analysis, asymptotic notation, Analyzing Control statement, Loop invariant and the correctness of the algorithm.

UNIT II MATHEMATICAL ASPECTS AND ANALYSIS OF ALGORITHM

Mathematical analysis of non- recursive algorithm , mathematical analysis of recursive algorithm, example: fibonacci numbers, empirical analysis of algorithms, algorithm visualization

UNIT III ANALYSIS OF SORTING AND SEARCHING ALGORITHM

Sorting Algorithms and Analysis: Bubble sort, Selection sort, Insertion sort,Shell sort Heap sort, Sorting in linear time: Bucket sort, Radix sort and Counting sort. sequential search and brute-force string matching, divide and conquer, merge sort, binary search, binary tree, traversal and related properties, depth first search and breadth forst search.

UNIT IV ALGORITHM TECHNIQUES

Transform and conquer, presorting, balanced search trees, avl trees, heaps and heap sort, dynamic programming, Warshall's and Floyd's algorithm, optimal binary search trees, greedy techniques, Prim's algorithm, Kruskal's algorithm, Dijkstra's algorithm, Huffman trees.

UNIT V ALGORITHM DESIGN METHODS

Backtracking, n-Queen's problem, Hamiltonian circuit problem, subset-sum problem, branch and bound, assignment problem, knapsack problem, traveling salesman problem.

Text Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithm", Pearson Education Asia, 2003
2. T.H. Cormen, C.E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithm", PHI Pvt. Ltd., 2001

Reference books:

1. Sara Baase and Allen Van Gelder,"Computer Algorithms-Introduction to the Design and Analysis ", Pearson Education Asia, 2003
2. A. V. Aho, J.E. Hopcroft and J.D. Ullman, "the Design and Analysis of Computer Algorithms", Pearson Education Asia, 2003.

DATA STRUCTURES			
Course Code:	**MAI104**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES

1. To emphasize the importance of appropriate data structure in developing and implementing efficient algorithms
2. Understand basic data structures such as arrays, stacks, queues, hash tables and linked list
3. To analyze the asymptotic performance of various algorithms
4. Solve problems using graphs, trees and heaps
5. Apply important algorithmic design paradigms and methods of analysis

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Define basic static and dynamic data structures and relevant standard algorithms for them.
2. Select basic data structures and algorithms for autonomous realization of simple programs or program parts.
3. Determine and demonstrate bugs in program, recognise needed basic operations with data structures
4. Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures
5. Evaluate algorithms and data structures in terms of time and memory complexity of basic operations

UNIT I INTRODUCTION TO DATA STRUCTURES

Abstract data types, sequences as value definitions, data types in C, pointers in C, data structures and C, arrays in C, array as ADT, one dimensional array, Implementing one dimensional array, array as parameters, two dimensional array, structures in C, implementing structures, Unions in C, implementation of unions, structure parameters, allocation of storage and scope of variables, recursive definition and processes: factorial function, fibonacci sequence, recursion in C, efficiency of recursion, hashing: hash function, open hashing, closed hashing: linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT II STACK, QUEUE AND LINKED LIST

Stack definition and examples, primitive operations, example -representing stacks in C, push and pop operation implementation, queue as ADT, C Implementation of queues, insert operation, priority queue, array implementation of priority queue, inserting and removing nodes from a list-linked implementation of stack, queue and priority queue, other list structures, circular lists: stack and queue as circular list - primitive operations on circular lists, header nodes, doubly linked lists, addition of long positive integers on circular and doubly linked list.

UNIT III TREES

Binary trees: operations on binary trees, applications of binary trees, binary tree representation, node representation of binary trees, implicit array representation of binary tree, binary tree traversal in C, threaded binary tree, representing list as binary tree, finding the Kth element, deleting an element, trees and their applications: C representation of trees, tree traversals, evaluating an expression tree, constructing a tree.

UNIT IV SORTING AND SEARCHING

General background of sorting: efficiency considerations, notations, efficiency of sorting, exchange sorts: bubble sort; quick sort; selection sort; binary tree sort; heap sort, heap as a priority queue, sorting using a heap, heap sort procedure, insertion sorts: simple insertion, shell sort, address calculation sort, merge sort, radix sort, sequential search: indexed sequential search, binary search, interpolation search.

UNIT V GRAPHS

Application of graph, C representation of graphs, transitive closure, Warshall's algorithm, shortest path algorithm, linked representation of graphs, Dijkstra's algorithm, graph traversal, traversal methods for graphs, spanning forests, undirected graph and their traversals, depth first traversal, application of depth first traversal, efficiency of depth first traversal, breadth first traversal, minimum spanning tree, Kruskal's algorithm, round robin algorithm.

Text Books:

1. Aaron M. Tenenbaum, Yeedidyah Langsam, Moshe J. Augenstein, 'Data structures using C', Pearson Education, 2004 / PHI.
2. E. Balagurusamy, 'Programming in Ansi C', Second Edition, TMH, 2003.

Reference books:

1. Robert L. Kruse, Bruce P. Leung Clovis L.Tondo, 'Data Structures and Program Design in C', Pearson Education, 2000 / PHI.

MACHINE LEARNING			
Course Code:	**MAI106**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/ P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45+00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Understand the foundational theory of Machine Learning:			
2. Formulate Machine Learning Problems:			
3. Master Various Algorithms and Their Limitations:			
4. Solve Problems of Moderate Complexity:			
5. Apply and Optimize Models:			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Design and Implement ML Solutions:			
2. Analyze Algorithm Performance:			
3. Apply ML Techniques to Real-World Problems:			
4. Preprocess and Understand Data:			
5. Develop Intelligent Systems:			

UNIT I INTRODUCTION

What is Machine Learning, Types of Machine Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Applications of Machine Learning – Stock Price Prediction, Face Recognition, Handwriting Recognition, Image Recognition, Virtual Personal Assistants, Medical Diagnosis, Online Fraud Detection.

UNIT II SUPERVISED LEARNING (REGRESSION/CLASSIFICATION)

Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods.

UNIT III UNSUPERVISED LEARNING

Clustering: K-means/Kernel K-means, Self-Organizing Maps. Dimensionality Reduction: PCA and kernel PCA Matrix Factorization and Matrix Completion Generative Models (mixture models and latent factor models).

UNIT IV ARTIFICIAL NEURAL NETWORKS

Biological Neurons and Biological Neural Networks, Artificial Neural Network, Types of Neural Network, Perceptron, History behind Perceptron, Importance of Perceptron, Working of Perceptron, Perceptron Learning, Percetron Learning Rule, Perceptron Learning of AND & OR gate, XOR gate, Activation functions, Binary Activation function, ReLU, Sigmoid, Hyperbolic, Softamax Activation function, Multilayer Perceptrons, Back propagation Neural Networks, and Feed-Forward Neural Networks. Applications and Future of Neural Networks.

UNIT V SELECTED TOPICS

Ensemble Methods (Boosting, Bagging, Random Forests), Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning, Recent trends in various learning techniques of machine learning and classification methods, Case studies in interdisciplinary domains.

Text Books:

1. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow by AurélienGéron, O'Reilly publication
2. An Introduction to Statistical Learning with Applications in R by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer publication (springer.com)

Reference Book:

1. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit- learn, and Tensor Flow 2, Publisher: Packt Publishing (December 12, 2019), Language: English, ISBN-10: 1789955750, ISBN-13: 978-1789955750
2. Machine Learning: The Absolute Complete Beginner's Guide to Learn and Understand Machine Learning From Beginners, Intermediate, Advanced, To Expert Concepts by Steven Samelson Publisher: Independently published (May 5, 2019)Language: English,ISBN-10: 1096853205,ISBN-13: 978-109685320

DATABASE MANAGEMENT SYSTEM			
Course Code:	**MAI108**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/P):**	**2P**
No. of Lectures + Tutorials (Hrs./Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L+T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Provide a strong foundation in basic database concepts, terminology, and applications.			
2. Teach Entity-Relationship (ER) modeling and relational model principles for designing databases to meet user requirements.			
3. Equip students with the ability to write effective Structured Query Language (SQL) statements for data definition, manipulation, and retrieval.			
4. Introduce database normalization techniques to optimize database design for data integrity and efficiency.			
5. Explain the concepts of transaction processing, concurrency control, and recovery for reliable database management.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Ability to analyze user requirements and design logical ER diagrams and relational schemas for databases.			
2. Ability to construct and execute complex SQL queries to perform data operations in relational databases.			
3. Ability to apply normalization techniques to normalize database schemas, improving data integrity and reducing redundancy			
4. Understanding and ability to apply principles of transaction management and concurrency control.			
5. Familiarity with database storage structures and access techniques, including indexing and hashing, to optimize performance.			

UNIT I DATA BASE SYSTEM

Data base system vs. file system, view of data, data abstraction, instances and schemas, data models, ER model, relational model, database languages, DDL, DML, database access for applications programs, data base users and administrator, transaction management, data base system structure, storage manager, query processor, history of data base systems, data base design and ER diagrams, beyond ER design entities, attributes and entity sets, relationships and relationship sets, additional features of ER model, concept design with the ER model, and conceptual design for large enterprises.

UNIT II RELATIONAL DATA BASE MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical database design, destroying /altering tables and views. relational algebra and calculus: relational algebra, selection and projection set operations, renaming, joins, division, relational calculus, tuple relational calculus, domain relational calculus, expressive power of algebra and calculus.

UNIT III SQL QUERY

Examples of basic SQL queries, nested queries, correlated nested queries set, comparison operators, aggregative operators, NULL values, comparison using null values, logical connectivity's, AND, OR and NOTR, impact on SQL constructs, outer joins, disallowing NULL values, complex integrity constraints in SQL triggers and active data bases.

UNIT IV NORMAL FORM

Problems caused by redundancy, decompositions, problem related to decomposition, reasoning about FDS, FIRST, SECOND, THIRD normal form, BCNF, forth normal form, fifth normal form, lossless join decomposition, dependency preserving decomposition, schema refinement in data base design, multi valued dependencies.

UNIT V TRANSACTION MANAGEMENT

ACID properties, transactions and schedules, concurrent execution of transaction, lock based concurrency control, performance locking, and transaction support in SQL, crash recovery, concurrency control, Serializability and recoverability, lock management, lock conversions, dealing with dead locks, specialized locking techniques, concurrency without locking, crash recovery: ARIES, log, other recovery related structures, the write, ahead log protocol, check pointing, recovering from a system crash, media recovery, other approaches and interaction with concurrency control.

Text Books:

1. Elmasri Navrate, Data Base Management System, Pearson Education, 2008.
2. Raghurama Krishnan, Johannes Gehrke, Data Base Management Systems, TMH, 3rd edition, 2008.
3. C. J. Date, Introduction to Database Systems, Pearson Education, 2009.

Reference books:

1. Silberschatz, Korth, Database System Concepts, McGraw hill, 5th edition, 2005.
2. Rob, Coronel & Thomson, Database Systems Design: Implementation and Management, 2009.

NATURAL LANGUAGE PROCESSING			
Course Code:	**MAI110**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/ P):**	**2P**
No. of Lectures + Tutorials (Hrs./Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. To Understanding the basics of natural language processing and understand various steps in it.			
2. To introduce the fundamentals of language processing from the algorithmic viewpoint.			
3. To discuss various issues that make natural language processing a hard task.			
4. Understand the importance and need of information retrieval system.			
5. To discuss some well-known applications of natural language processing.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Appreciate the fundamental concepts of natural language processing.			
2. Design algorithms for natural language processing tasks.			
3. Develop useful systems for language processing and related tasks involving text processing.			
4. Learn about machine translation.			
5. Ability to perform independent research and analysis.			

UNIT I INTRODUCTION

Natural Language Processing tasks in syntax, semantics, and pragmatics – Issues – Applications – The role of machine learning – Probability Basics –Information theory – Collocations -N-gram Language Models -Estimating parameters and smoothing – Evaluating language models.

UNIT II WORD LEVEL AND SYNTACTIC ANALYSIS

Word Level Analysis: Regular Expressions-Finite-State Automata-Morphological Parsing- Spelling Error Detection and Correction-Words and Word classes-Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar-Constituency- Parsing-Probabilistic Parsing.

UNIT III SEMANTIC ANALYSIS AND DISCOURSE PROCESSING

Semantic Analysis: Meaning Representation-Lexical Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: Cohesion-Reference Resolution- Discourse Coherence and Structure.

UNIT IV NATURAL LANGUAGE GENERATION AND MACHINE TRANSLATION

Natural Language Generation: Architecture of NLG Systems- Generation Tasks and Representations Application of NLG. Machine Translation: Problems in Machine Translation- Characteristics of Indian Languages- Machine Translation Approaches- Translation involving Indian Languages.

UNIT V INFORMATION RETRIEVAL AND LEXICAL RESOURCES

Information Retrieval: Design features of Information Retrieval Systems-Classical, Non- classical, Alternative Models of Information Retrieval – valuation Lexical Resources: WorldNet-Frame Net-Stemmers-POS Tagger- Research Corpora.

Text Books:

1. Daniel Jurafsky , James H. Martin , "Speech & language processing", Pearson publications.
2. Allen, James. Natural language understanding. Pearson, 1995.
3. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, "NLP: A Paninian Perspective", Prentice Hall, New Delhi.

Reference book:

1. L. M. Ivansca, S. C. Shapiro, "Natural Language Processing and Language Representation", AAAI Press, 2000.
2. T. Winograd, Language as a Cognitive Process, Addison-Wesley.

THEORY OF COMPUTATION			
Course Code:	**MAI112**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U/ P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45+00**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Understand Abstract Models of Computation			
2. Analyze Formal Language Hierarchies			
3. Comprehend Decidability and Undecidability			
4. Apply Mathematical Proof Techniques			
5. Apply Mathematical Proof Techniques			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Students will be able to design finite automata (DFA, NFA) and pushdown automata (PDA) for specified regular and context-free languages.			
2. Students will be able to construct context-free grammars to describe formal languages and understand different normal forms.			
3. Students will be able to design Turing Machines and understand their power to solve computational problems.			
4. Students will be able to differentiate between decidable and undecidable problems and understand the significance of computability.			
5. Students will gain foundational knowledge essential for advanced topics in computer science, such as compilers, artificial intelligence, and formal verification			

UNIT I INTRODUCTION

Introduction; alphabets, strings and languages; automata and grammars, deterministic finite automata (DFA)-formal definition, simplified notation: state transition graph, transition table, language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, language of NFA, equivalence of NFA and DFA, minimization of finite automata, distinguishing one string from other, Myhill-Nerode Theorem.

UNIT II REGULAR EXPRESSIONS

Regular expression (RE), definition, operators of regular expression and their precedence, algebraic laws for regular expressions, Kleen"s theorem, regular expression to FA, DFA to regular expression, arden theorem, non regular languages, pumping lemma for regular languages. application of pumping lemma, closure properties of regular languages, decision properties of regular languages, FA with output: moore and mealy machine, equivalence of moore and mealy machine, applications and limitation of FA.

UNIT III CFG

Context Free Grammar (CFG) and Context Free Languages (CFL): definition, examples, derivation, derivation trees, ambiguity in grammar, inherent ambiguity, ambiguous to unambiguous CFG, useless symbols, simplification of CFGs, normal forms for CFGs: CNF and GNF, closure properties of CFLs, decision properties of CFLs: emptiness, finiteness and membership, pumping lemma for CFLs.

UNIT IV PUSH DOWN AUTOMATA

Push Down Automata (PDA): description and definition, instantaneous description, language of PDA, acceptance by final state, acceptance by empty stack, deterministic PDA, equivalence of PDA and CFG, CFG to PDA and PDA to CFG, two stack PDA.

UNIT V TURING MACHINES (TM)

Basic model, definition and representation, instantaneous description, language acceptance by TM, variants of turing machine, TM as computer of integer functions, universal TM, church"s thesis recursive and recursively enumerable languages, halting problem, introduction to undecidability, undecidable problems about TMs. Post Correspondence Problem (PCP), modified PCP, introduction to recursive function theory.

Text Books:

1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education
2. K.L.P. Mishra and N.Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", PHI

Reference books:

1. Martin J. C., "Introduction to Languages and Theory of Computations", TMH
2. Papadimitrou, C. and Lewis, C.L., "Elements of the Theory of Computation", PHI

MACHINE LEARNING LAB			
Course Code:	**MAI84**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES

1. To introduce students to the basic concepts and techniques of Machine Learning
2. A general understanding of ML process models.
3. To introduce students to the basic concepts and techniques of Machine Learning
4. Understanding of Python Programming and its module
5. Understanding of Deep learning.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Installation of python and its module &ipython notebook.
2. Ability to apply Python principles and techniques.
3. Ability to design ML models and test and train data set .
4. To Understand working of tensorflow.
5. Ability to perform deep learning algorithms.

LIST OF PRACTICALS

1. Installation of Python and python Notebook.
2. Implement- Data Types and Containers in Python.
3. A scatter plot is a diagram where each value in the data set is represented by a dot.
4. Implement Regression to find the relationship between variables.
5. Machine Learning - Train/Test- Evaluate Your Model
6. Implement polynomial regression - R-squared, Predict the future and Bad fit.
7. Implement - Machine Learning - Decision Tree.
8. Install - Python MySQL,MySQLDatabase,Install MySQL Driver,Test MySQL Connector,Create Connection
9. Introduction to Deep Learning - Deep Learning basics with Python, TensorFlow and Keras p.1
10. Optimizing Models with TensorBoard - Deep Learning basics with Python, TensorFlow.

DATA BASE MANAGEMENT SYSTEM LAB			
Course Code:	**MAI186**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES

1. Understand the basics and functions of MS excel.
2. Clear understanding and use of data validations and templates.
3. Purpose of sorting and filtering features.
4. Use of reports in business organizations.
1. Purpose and advantage of charts for top management in any work place.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Learn to understand the functions in Excel.
2. Understand the validations.
3. Make reports in excel.
4. Learn to work with pivot tables.
5. Learn how to make charts in MS excel.

LIST OF EXPERIMENTS:

1. Write the queries for Data Manipulation and Data Definition Language.
2. Write SQL queries using logical operations and operators.
3. Write SQL query using group by function.
4. Write SQL queries for sub queries, nested queries
5. Write SQL queries to create views.
6. Write an SQL query to implement JOINS.
7. Write a query for extracting data from more than one table.
8. Write a query to understand the concepts for ROLL BACK, COMMIT & CHECK POINTS.
9. Create tables according to the following definition.
    a. Create table deposit (actno varchar2(5) ,cname varchar2(18) , bname varchar2(18) , amount number(8,2) ,adate date);
    b. Create table branch(bname varchar2(18),city varchar2(18));
    c. Create table customers(cname varchar2(19) ,city varchar2(18));
    d. Create table borrow(loanno varchar2(5), cname varchar2(18), bname varchar2(18), amount number (8,2));
10. Retrieve all data from employee, jobs and deposit.
    a. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06
    b. Display all jobs with minimum salary is greater than 4000.
    c. Display name and salary of employee whose department no is 20. Give alias name to name of employee.
    d. Display employee no,name and department details of those employee whose department lies in(10,20)

Natural Language Processing Lab			
Course Code:	**MAI 188**	**COURSE CREDITS:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Labs(Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**-**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3 Hrs**
COURSE OBJECTIVES			
1. Explain basic natural language processing concepts, applications and interface models with real life.			
2. Demonstrate the use of natural language with computer networks.			
3. Emphasize the importance of python language in NLP.			
4. Facilitate students in NLP for the modern age.			
5. Familiarize issues of NLP with system interface.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Students get practical knowledge on designing and creating relational NLP based systems.			
2. Understand various advanced queries and execute NLP with Python language.			
3. Students are able to implement several NLP applications.			
4. Students are able to implement several NLP applications.			

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments:

1. Write a python program to perform tokenization by word and sentence using nltk.
2. Write a python program to eliminate stop words using nltk.
3. Write a python program to perform Parts of Speech tagging using nltk.
4. Write a python program to perform lemmatization using nltk.
5. Write a python program for chunking using nltk.
6. Write a python program to find the probability of the given statement "This is my cat" by taking the example corpus into consideration.
7. Write the python code to perform sentiment analysis using NLP
8. Write the python code to detect Fake News using NLP.
9. Write a python program to find Term Frequency and Inverse Document Frequency (TF-IDF).
10. Write the python code to develop Spam Filter using NLP.

PROFESSIONAL ETHICS			
Course Code:	**ITV301**	**Course Credits:**	**1**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures + Tutorials (Hrs/Week):	**02 + 00**	**Mid Sem. Exam Hours:**	**1.5 Hrs**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3 Hrs**

COURSE OBJECTIVES
1. Describe the Professional ethics and morals
2. Explain the basic concepts of Global business Ethics
3. Explain the ethical theory
4. Workplace ethics in changing domain of research
5. Improve the safety and moral leadership
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand How to live life peacefully by following Professional ethics
2. Social responsibilities of Global business
3. Understand Emotional Intelligence as Social Experimentation.
4. Execute Safety, Responsibilities and Rights
5. Explain the principle of Professional ethics and Environmental responsibility.

UNIT I Introduction:

Individual and Professional Ethics Introduction to Professional Ethics, Morals, Values and Ethics. Personal and Professional sensé towards society,Code of Ethics by Morals,values. Integrity,Right of equality, Academic integrity,Work Ethics,Society Service ,Respect for others, Living peacefully, Caring and Sharing,Honesty, courage,Cooperation commitment,Empathy,Self Confidence,Social Expectations.

UNIT II Business Ethics

Philosophical approaches to Business Ethics, ethical reasoning,ethical issues in business,Social Responsibility of Business conflict of interest,cultural relativism,Ethical leadership,Resisting un-ethical authority and domination,Global Business Ethics, Models of professional roles,Theories about right action,Self interest,Customs and Religion,Uses of Ethical Theories.

UNIT III Psychological Approaches

Ethical Theories,Psychological and Philosophical approaches, Myths about Morality,conflict of interest in psychological perspective,Courage, Integrity, Emotional Intelligence as social Experimentation, Plagiarism, A balanced outlook on law,Challenges case study,Bhopal gas tragedy.

UNIT IV Workplace Ethics

Ethics in changing domains of Research, academic integrity, intellectual honesty, Role of Engineers and Managers, Ethical issues in Diverse workplace, competition, free will Confidentiality, employee rights, Intellectual property rights, discrimination Responsibilities and Rights Role of confidentiality in moral integrity, Conflicts of interest, Occupational crime Professional rights Employee right

UNIT V Safety, Responsibilities and Rights

Ecology, Engineering, Economy,Risk benefit analysis,Corporate social responsibility and Corporate Sustainability,CSR in India, Sustainability, Case Studies, Global Ethical Issues. Environmental Ethics, Business Ethics,Computer Ethics,Role in Technological Development and Moral leadership

Text Books:

1. Dr.B.R. Ambedkar (1979). Dr. Babasaheb Ambedkar, writings and speeches, Volume 1. Education Dept.,Govt.of Maharashtra ,Pp.33-87.
2. Subramanian.R. Professional Ethics, Oxford Publication, 2013.
3. Nagarasan. R.S. Professional Ethics and Human Values. New Age International Publications, 2006.
4. Mike W Martin and Roland Schinzinger, Ethics in Engineering,4th edition, Tata McGraw Hill Publishing Company Pvt Ltd, New Delhi,2014

Reference Books:

1. Mike W Martin and Roland Schinzinger, Ethics in Engineering,4th edition, Tata McGraw Hill Publishing Company Pvt Ltd, New Delhi,2014.
2. Charles D Fleddermann, Engineering Ethics, Pearson Education/ Prentice Hall of India, New Jersey,2004.
3. Charles E Harris, Michael S Protchard and Michael J Rabins, Engineering Ethics- Concepts and cases, Wadsworth Thompson Learning, United states,2005.