UNIVERSITY OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE STRUCTURE

B.TECH. COMPUTER SCIENCE AND ENGINEERING SPECIALIZATION IN AI

2022-2026

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

SEMESTER I

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA101	Engineering Mathematics-I	3	1	0	4	GE1
2	PH102	Engineering Physics	3	1	0	4	GE2
3	CS101	Fundamentals of Computer Programming	3	1	0	4	CC1 / FC
4	CS105	Introduction of Artificial Intelligence	2	0	0	2	CC2 / FC
5	EC101	Basic Electronics Engineering	3	1	0	4	GE3 / FC
6	EN101	English Proficiency	2	0	0	2	OE1 / AECC
7	CE103	Engineering Graphics Lab	1	0	2	2	GE-L1
8	PH104	Engineering Physics Lab	0	0	2	1	GE-L2
9	CS181	Computer Programming Lab	0	0	2	1	CC-L1 / SEC
10	EC181	Basic Electronics Engineering Lab	0	0	2	1	GE-L3
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**17**	**4**	**8**	**25**	

SEMESTER II

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA102	Engineering Mathematics-II	3	1	0	4	GE4
2	AI102	Introduction to Python	2	0	0	2	CC3 / FC
3	EE102	Basic Electrical Engineering	3	1	0	4	GE5
4	ME101	Engineering Mechanics	3	1	0	4	GE6
5	ES101	Environmental Studies	3	1	0	4	OE2 / AECC
6	AI104	Data Structure and Algorithm	2	0	0	2	CC4 / SEC
7	AI182	Data Structure and Algorithm Lab	0	0	2	1	CC-L2 / SEC
8	EE104	Basic Electrical Engineering Lab	0	0	2	1	GE-L4
9	ME102	Workshop Practice	1	0	2	2	GE-L5
10	AI184	Python Lab	0	0	2	1	CC-L3 / SEC
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**17**	**4**	**8**	**25**	

SEMESTER III

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	AI201	Software Engineering	3	0	0	3	CC5
2	AI203	Intelligent Systems	3	0	0	3	CC6
3	AI205	Theory of Computations and Theory of automata	3	0	0	3	CC7
4	AI207	Database Management Systems	3	0	0	3	CC8
5	AI209	Image processing and computer vision	3	0	0	3	CC9
6	AI211	Introduction to R Programming	3	1	0	4	CC10 / SEC
7	AI281	Database Management Systems Lab	0	0	3	2	CC-L4
8	AI283	Image Processing and Computer Vision Lab	0	0	3	2	CC-L5
9	AI285	R Programming Lab	0	0	3	2	CC-L6 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER IV

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	AI202	Machine Learning	3	1	0	4	CC11
2	AI204	Operating System	3	0	0	3	CC12
3	AI206	Computer Interfacing and Embedded Systems	3	0	0	3	CC13
4	AI208	Design and Analysis of Algorithms	3	0	0	3	CC14 / SEC
5	AI210	Quantum Computing	3	0	0	3	CC15
6	AI212	Computer Networks	3	0	0	3	CC16
7	AI282	Machine Learning using Python Lab	0	0	3	2	CC-L7 / SEC
8	AI284	Operating System Lab	0	0	3	2	CC-L8
9	AI286	Design and Analysis of Algorithms Lab	0	0	3	2	CC-L9 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER V

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	AI301	Big Data Analytics	3	0	0	3	CC17
2	AI303	Compiler Design	3	0	0	3	CC18
3	AI305	Deep Learning and Reinforcement Learning	3	0	0	3	CC19 / SEC
4	AI307	Operational Information Security Management and Biometrics	3	1	0	4	CC20
5		Elective 1	3	0	0	3	E1 / DSE
6		Elective 2	3	0	0	3	E2 / DSE
7	AI381	Big Data Analytics Lab	0	0	3	2	CC-L10 / SEC
8	AI383	Compiler Design Lab	0	0	3	2	CC-L11
9	AI385	Deep Learning Lab using Python	0	0	3	2	CC-L12
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER VI

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	AI302	IoT and Its Applications	3	0	0	3	CC21
2	AI304	Expert Systems	3	0	0	3	CC22
3	AI306	Cloud Computing	3	0	0	3	CC23
4	AI308	Metaheuristics for optimization	3	1	0	4	CC24
5		Elective 3	3	0	0	3	E3 / DSE
6		Elective 4	3	0	0	3	E4 / DSE
7	AI382	Internet of Things Lab	0	0	3	2	CC-L13
8	AI384	Expert Systems Lab	0	0	3	2	CC-L14 / SEC
9	AI386	Cloud Computing Lab	0	0	3	2	CC-L15
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

Industrial Training will be done by candidate individually after third year during the summer break and it will be ofminimum 4 weeks. It will be evaluated as per University Examination in VII semester.

SEMESTER VII

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA402	Parallel Processing and CUDA Programming	3	1	0	4	GE7
2	AI401	Computational Intelligence	3	0	0	3	CC25
3	AI403	Robotics and Drones	2	0	0	2	CC26
4	AI405	Natural Language Processing	3	0	0	3	CC27
5		Elective 5	3	0	0	3	E5 / DSE
6	AI481	Robotics and Drones Lab	0	0	3	2	CC-L16
7	AI491	Minor Project	0	0	6	3	IT1 / E
8	AI493	Industrial Training	0	0	10	5	MP1 / E
9	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**14**	**1**	**19**	**25**	

SEMESTER VIII

SR. 0.	Course Code	Course Name	L	T	P	Credits	Types
1	AI490	Seminar	0	0	3	2	S / E
2	AI492	Major Project	0	0	16	8	I / E
3	AI494	Internship	0	0	30	15	MP2 / E
4	GP	General Proficiency	Non Credit				
Total Hours and Credits			**0**	**0**	**49**	**25**	

In the **Seminar**, student need to study and present individually, on latest research paper of their specialized area and It will be evaluated as per University Examination Rules.

The **Internship** in Industry will be done by candidate individually during the 8th semester and it will be for a minimum of 4 (-6) months. It will be evaluated as per University Examination Rules.

Minor and Major Project will be in a group and It will be evaluated as per University

Examination Rules. USICT will provide a mentor/supervisor for industrial training,

seminar, internship, minor and major projects.

ELECTIVES FROM DCSE

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	AI309	Computer Graphics	3	0	0	3	E1
2	AI311	Introduction to Brain and Neuroscience	3	0	0	3	E1
3	AI313	Stochastic Processes	3	0	0	3	E1
4	AI315	Computer Based Numerical & Statistical Techniques	3	0	0	3	E1
5	AI317	Sequence Models	3	0	0	3	E1
6	AI319	Bayesian Data Theory	3	0	0	3	E2
7	AI321	Speech Analysis and Systems	3	0	0	3	E2
8	AI323	Graph Theory	3	0	0	3	E2
9	AI325	Distributed Database	3	0	0	3	E2
10	AI327	Pattern Recognition	3	0	0	3	E2
11	AI310	Biometric Security	3	0	0	3	E3
12	AI312	Gaming	3	0	0	3	E3
13	AI314	Knowledge Engineering	3	0	0	3	E3
14	AI316	Predictive Analysis	3	0	0	3	E3
15	AI318	Digital Fabrication	3	0	0	3	E3
16	AI320	AI Enabled Cyber Security	3	0	0	3	E4
17	AI322	Augumentad and Virtual Reality	3	0	0	3	E4
18	AI324	Fuzzy logic	3	0	0	3	E4

19	AI326	Distributed Operating Systems	3	0	0	3	E4
20	AI328	Business Analytics	3	0	0	3	E4
21	AI407	Automation and Robotics	3	0	0	3	E5
22	AI409	Blockchain Technology using SALONA	3	0	0	3	E5
23	AI411	3D Printing	3	0	0	3	E5
24	AI413	Parallel Distributed Systems	3	0	0	3	E5
25	AI415	Time Series Analysis and Applications	3	0	0	3	E5

SEMESTER I

FUNDAMENTALS OF COMPUTER PROGRAMMING			
Course Code:	**CS101**	**Course Credits:**	**4**
Course Category:	**C**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 02**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. The major objective is to provide students with understanding of programming essentials and to learn preprogramming steps like writing algorithms, drawing flowcharts and pseudo codes.
2. Understand the structure, and learn the syntax and semantics of C programming
3. Understand variable declaration with different data types and learn using operators and different control structures like decision control, loop control and special cases..
4. Understanding the concept of pointers, declarations, initialization, operations on pointers and their usage
5. Understanding how to perform various FILE I/O.

COURSE OUTCOMES

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

1. Able to implement the algorithms and draw flowcharts for solving Mathematical and Engineering problems.
2. Students can write, compile and debug programs in C language and use different data types for writing the programs
3. Able to design programs connecting decision structures, loops and functions.
4. Understand the dynamic behavior of memory by the use of pointers.
5. Develop confidence for self-education and ability for life-long learning needed for Computer language.

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, compil- er, interpreter and assembler, linker, loader, debugger, IDE. Introduction to algorithm and flow chart; representation of algorithm using flow chart 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, Pre-defined 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, dy-namic 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. C Programming, Herbert Shield.
2. C Programming Language 2nd Edition by Brian, W Kernighan Pearson Education.

Reference Books:

1. Programming in ANSI C by E. Balagurusamy, Tata Mgraw Hill.
2. C Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer Prentice Hall- Gale.
3. Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling Kindersley India.
4. Introduction To UNIX System by Morgan Rachel Tata Mcgraw Hill Education.
5. C: A Reference Manual (5th Edition) by Samuel P. Harbison&Samuel P. Harbison.
6. Programming Using the C Language by Hutchison,R.C, Mcgraw Hill Book Company, New York
7. Fundamentals of computers and programming with C, A.K. SHARMA

INTRODUCTION TO ARTIFICIAL INTELLIGENCE			
Course Code:	**CS105**	**Course Credits:**	**2**
Course Category:	**CC**	**Course(U/P)**	**U**
Course Year(U/P):	**1U**	**Course Semester(U/P):**	**1U**
No. of Lectures + Tutorials (Hrs/Week):	**02+00**	**Mid Sem . Exam Hours:**	**1.5**
Total No. of Lectures (L+T):	**30+00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES:			
1.Provide a strong foundation of fundamental concepts in Artificial Intelligence			
2.Enable the student to apply these techniques in applications which involve perception, reasoning and learning			
3.Provide a basic exposition to the goals and methods ofArtificial Intelligence			
4.Explain the role of agents and how it is related to environment and the way of evaluating it and how agents can act by establishing goals.			
5.Learn the different machine learning techniques to design AI machine and enveloping applications For real world problems.			
COURSE OUTCOMES:			
At the end of the course the students should be able to:			
1.Understand the various searching techniques, constraint satisfaction problem and example problems-game playing techniques.			
2.Apply these techniques in applications which involve perception, reasoning andlearning			
3.Acquire the knowledge of real world Knowledge representation			
4.Analyze and design a real world problem for implementation and understand the dynamic behavior of a system.			
5.To enable the student to apply these techniques in applications which involve perception, reasoning and learning			

UNIT 1 Introduction

Introduction to AI, Components of AI, Goals of AI, Types of AI, History of AI, Turing Test in AI, Intelligence, Advantages and Disadvantages of AI, Intelligent System, Role of IS, Comparison of various IS, Weak and strong AI, Mind-Body Problem in AI, Chinese Room Experiment in AI, Parallel and Distributed AI.

UNIT 2 Agents in AI

Intelligent Agents, Types of Agents: Simple Reflex Agent, Model-based reflex agent, Goal-based agents, Utility-based agent, Learning agent, Structure of Agents, Agent Environment in AI, Examples of Agents, Knowledge Engineering, Knowledge Based System, Knowledge Engineering Techniques, Knowledge Engineering Principle, Knowledge Engineering Methodologies.

UNIT 3 Searching Techniques and AI Problems

Searching in AI, Search Algorithm terminology, Properties of Search Algorithm, Breadth First Search, Depth First Search, Best First Search, Tic-Tac-Toe problem, Water-Jug Problem, Chess Problem, Tower of Hanoi Problem, Travelling Salesperson Problems, Monkey and Banana Problem, Magic Square.

UNIT 4 Knowledge Representation

Knowledge Representation definition, Declarative knowledge, Procedural Knowledge, Meta knowledge, Heuristic Knowledge, Structural knowledge, Inheritable Knowledge, Inferential Knowledge, Relational Knowledge, Explicit Knowledge, Tacit Knowledge, Uncertain Knowledge, Knowledge Storage, Relation between Knowledge and Intelligence, AI knowledge Cycle.

UNIT 5 AI Techniques and Applications

Introduction of Machine learning, Introduction of Deep Learning, Introduction to Expert system, Introduction to Natural Language Process, AI in future, AI in social media, AI in Entertainment and Education, AI in Drones, AI in Automated customer 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.

Reference Books:

1. Artificial Intelligence, Elanie Reich: Tata mcgraw Hill publishing house, 2008.
2. Artificial Intelligence, Ela Kumar, IK Publishing.
3. Artificial intelligence, Peterson, TataMcGraw Hill, 2008.
4. Artificial intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
5. Artificial Intelligence, Winston, PHI publication, 2006.
6. Artificial Intelligence – A Modern Approach (3rd Edition) By Stuart Russell & Peter Norvig
7. Artificial Intelligence: The Basics By Kevin Warwick

COMPUTER PROGRAMMING LAB – I			
Course Code:	**CS181**	**Course Credits:**	**1**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs/Week):	**02 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. To introduce students to the basic knowledge of programming fundamentals of C language.
2. To impart writing skill of C programming to the students and solving problems.
3. To impart the concepts like looping, array, functions, pointers, file, structure.
4. Write programs to print output on the screen as well as in the files.
5. Apply all the concepts that have been covered in the theory course.

COURSE OUTCOMES

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

1. Recognize and understand the syntax and construction of C programming code
2. Able to design and develop Computer programs, analyses, and interprets the concept of pointers, declarations, initialization, operations on pointers and their usage.
3. Able to define data types and use them in simple data processing applications also he/she must be able to use the concept of array of structures.
4. Student must be able to define union and enumeration user defined data types.
5. Develop confidence for self-education and ability for life-long learning needed for Computer language.

EXPERIMENTS:

1. Write a C program to reverse a given number, find the sum of digits of the number.
2. Write a C program to concatenate two strings.
3. Write a C program to take marks of a student as input and print the his/her grade bases on following criteria using if – else statements

   Marks <40 FAIL

   40<= Marks <59 GOOD

   59 <= Marks < 80 Excellent

   80 <= Marks Outstanding

4. Perform experiment 3 using switch case statement.

4. Write a C program to convert all the lowercase letter to uppercase letter and all uppercase letters to lower case letter given a string as input.

5.Write a C program to compute the roots of a quadratic equation.

6. Write a C program to check whether a given number is prime or not, also check whether it is divisible by a number k or not.

7. Write a C program to check whether a given year is leap year or not.

8. Write a C program to take two matrixes as input and print the sum of two matrixes.

9. Write a C program to display the address of a variable using pointer.

10. Write a C program to compute the length of a string using pointer.

11. Create a structure called STUDENT having name, registration number, class, session as its field. Compute the size of structure STUDENT

12. Write a C program to check weather a given string is palindrome or not.

13. Write a C program to generate following patterns.

```
    1
   2 2
  3 3 3
 4 4 4 4
```

```
    A
   B B
  C C C
 D D D D
```

SEMESTER II

INTRODUCTION TO PYTHON			
Course Code:	**AI102**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Lectures + Tutorials (Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**30**	**End Sem. Exam Hours:**	**3**
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 BASSICS, 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 Analysis-Numpy** variable, Numpy manipulation, Skippy, Pandas intro. Descriptive analysis, Pandas Input-output, Pandas manipulation, Pandas group by

Text Books:

Head First Python 2e: A Brain-Friendly Guide Paperback – Illustrated, 16 by Paul Barry, Oreilly

Python: The Complete Reference Paperback – 20 March 2018 by Martin C. Brown (Author), TMH Publication

Let Us Python by YashavantKanetkar , 1 January 2019, BPB publication

Python Programming, A modular approach , First Edition, By Pearson Publication by Taneja Sheetal and Kumar Naveen , 26 September 2017

DATA STRUCTURE AND ALGORITHMS			
Course Code:	**AI104**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Lectures + Tutorials (Hrs./Week):	**02 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**30 + 00**	**End Sem. Exam Hours:**	**3**

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, YeedidyahLangsam, Moshe J. Augenstein, 'Data structures using C', Pearson Education, 2004 / PHI.
2. E. Balagurusamy, 'Programming in Ansi C', Second Edition, TMH, 2003.
3. Robert L. Kruse, Bruce P. Leung Clovis L.Tondo, 'Data Structures and Program Design in C', Pearson Education, 2000 / PHI.

PYTHON LAB			
Course Code:	**AI184**	**Course Credits:**	**1**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Lectures + Tutorials (Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**10**	**End Sem. Exam Hours:**	**3**
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-elf-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, Pandas group by

Text Books:

Head First Python 2e: A Brain-Friendly Guide Paperback – Illustrated, 16 by Paul Barry, Oreilly

Python: The Complete Reference Paperback – 20 March 2018 by Martin C. Brown (Author), TMH Publication

Let Us Python by YashavantKanetkar , 1 January 2019, BPB publication

Python Programming, A modular approach , First Edition, By Pearson Publication by Taneja Sheetal and Kumar Naveen , 26 September 2017

DATA STRUCTURE AND ALGORITHMS LAB			
Course Code:	**AI182**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Labs (Hrs/Week):	**2**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

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.
4. To develop application using data structure algorithms
5. 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 advanced 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

9. Implementation of Hash Table

SEMESTER III

SOFTWARE ENGINEERING			
Course Code:	**AI201**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

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 design, 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.

REFERENCE 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.

INTELLIGENT SYSTEMS			
Course Code:	**AI203**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Demonstrate good knowledge of basic theoretical foundations of Artificial Intelligence and intelligent systems.
2 Representing a problem as a search solving problem.
3 Logical representations of uncertainty, and rational decision making in uncertain environments.
4 Provide understanding of techniques, mathematical concepts, and algorithms used in machine learning to facilitate further study in this area
5 Study the fundamental algorithms for pattern recognition

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Students will be enabled to understand nature and objective of basic foundations of the intelligent systems methodologies such as Case-based reasoning,Intelligent agents
2 Students will determine which type of intelligent system methodology would be suitable for a given type of application problem & demonstrate in the form of a major project work.
3 . Students will able to design and develop an intelligent system for a selected application.
4 Understand a very broad collection of machine learning algorithms and problems.
5 Understand and apply various algorithms for pattern recognition

UNIT I INTRODUCTION

Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents. Computer vision, Natural Language Possessing.

UNIT II INTRODUCTION TO SEARCH

Searching for solutions, Uniformed search strategies, Informed search strategies, Local search algorithms and optimistic problems, Adversarial Search, Search for games, Alpha - Beta pruning.

UNIT III KNOWLEDGE REPRESENTATION & REASONING

Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.

UNIT IV MACHINE LEARNING

Supervised and unsupervised learning, Decision trees, Statistical learning models, Learning with complete data – Naïve Bayes models, Learning with hidden data – EM algorithm, Reinforcement learning

UNIT V PATTERN RECOGNITION

Introduction, Design principles of pattern recognition system, Statistical Pattern recognition, Parameter estimation methods - Principle Component Analysis (PCA) and Linear Discriminate Analysis (LDA), Classification Techniques – Nearest Neighbour (NN) Rule, Bayes Classifier, Support Vector Machine (SVM), K – means clustering.

Text Books

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

2. Artificial intelligence, Peterson, TataMcGraw Hill, 2008.

3. Artificial intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.

4. Artificial Intelligence, Winston, PHI publication, 2006.

Theory of Computation and Theory of Automata			
Course Code	**AI205**	**Course Credit**	**03**
Course Category	**CC**	**Course(U/P)**	**U**
Course year	**2U**	**Course Semester(U/P)**	**3U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**45+00**	**End Term Exam Hours:**	**03**
COURSE OBJECTIVES			
1. The objective of this course is to introduce students to the foundation of computability theory.			
2. Application of mathematical techniques and logical reasoning to important problem.			
3. Develop a strong background in reasoning about finite state automata and formal language.			
4. This course is to explore the theoretical foundations of computer science from the perspective of formal language and classify machines by their power to recognize languages.			
5. the basic theory of computer science and formal methods of computation like automation theory, formal language, grammars, Turing machine			
Course Outcomes			
At the end of the course the student should be able to understand the :			
1. Under the basic property of regular grammar and design automata			
2.Language accepted by an automata i.e. DFA(Deterministic Finite Automata)/NDFA(Non deterministic finite automata).			
3. Understand the regular expression(RE) ,Kleen closure ,positive closure, RE to FA and FA to RE			
4.Closure property of different language and Decidability /Undesirability property of different languages.			
5.Define the various categories of language grammars in the Chomsky hierarchy and variants of Turing machine			

UNIT I 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, Quotient Construction, My hill- Nerode Theorem.

UNIT II Regular expression

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 proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs Cock-Younger-Kasami Algorithm, Application to Parsing.

UNIT V Turing machines

Turing machies (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 vs. Looping, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory .

Text Books

1.Automata and Computability, Dexter C. Kozen , Springer Publishers, 2007.
2. Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani, and Ullman, Pearson Publishers, Third Edition, 2006.

Reference Books

1.Elements of the Theory of Computation, H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers, 1981
2.Introduction to Languages and the Theory of Computation, John. C. Martin, Tata McGraw-Hill, 2003.
3.K.L.P. Mishra and N.Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", PHI Learning Private Limited, Delhi India

Database Management Systems			
Course Code	**AI207**	**Course Credit**	**03**
Course Category	**CC**	**Course(U/P)**	**U**
Course year(U/P)	**2U**	**Course Semester(U/P)**	**3U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**45+00**	**End Term Exam Hours:**	**03**
COURSE OBJECTIVES			
1. The objective of this course is to introduce students to the foundation of DBMS			
2. Application of DBMS			
3.How to organize, maintain, retrieve efficiently and effectively- information from a DBMS.			
4. This course is to explore the practical foundations of DBMS.			
5. The basic theory DBMS and practical methods of DBMS like organize maintain retrieve information from DBMS.			
Course Outcomes			
At the end of the course the student should be able to understand the :			
1. . Students will have an understanding of database concepts and thorough knowledge of database software.			
2. Students will be able to model an application's data requirements using ER diagrams			
3. Students will be able to write SQL commands to create tables and query data in a relational DBMS			
4. Students will be able to execute various advance SQL queries related to transactions, concurrency			
5.Students will be able to Transactions of DBMS			

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 MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical data base 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 BASIC 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 SCHEMA REFINEMENT

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

UNIT V OVERVIEW OF 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.

References Books:

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

Image Processing and Computer Vision			
Course Code:	**AI 209**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Computer Vision focuses on development of algorithms and techniques to analyze and interpret the visible world around us.
2 This requires understanding of the fundamental concepts related to multi-dimensional signal processing, feature extraction, pattern analysis visual geometric modeling, stochastic optimization etc.
3 Focus on early processing of images and the determination of structure: edges, lines, shapes
4Aims at calculating the structure and depth of objects in a scene from a set of multiple views or images..
5Understanding the various image segmentation and feature extraction methods .
COURSE OUTCOMES
At the end of the course the students should be able to:
1Identify basic concepts, terminology, theories, models and methods in the field of computer vision.
2Describe basic methods of computer vision related to multi-scale representation, edge detection and detection of other primitives, stereo, motion and object recognition.
3Assess methods to use for solving a given problem, and analyse the accuracy of the methods .
4Recognize the object using the concept of computer vision K4
5Detect a moving object in video using the concept of motion analysis K3.

UNITI Digital Image Formation and low-levelprocessing

Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement Restoration, Histogram Processing.

UNITII Depth estimation and Multi-cameraviews

Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homograph, Rectification, DLT,RANSAC, 3- D reconstruction framework; Auto-calibration.

UNITIII Feature Extraction and Image Segmentation

Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space

Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT, Region Growing, Edge Based approaches to segmenation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

UNITIV Pattern and MotionAnalysis

Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un- supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods, Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation

UNITV Shape fromX

Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

Text Books

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited2011.
2. ComputerVision:AModernApproach,D.A.Forsyth,J.Ponce,PearsonEducation,2003.

Reference Books

1. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March2004.
2. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley,1992.
3. K.Fukunaga;IntroductiontoStatisticalPatternRecognition,SecondEdition,Academic Press, MorganKaufmann,1990

INTRODUCTION TO R PROGRAMMING			
Course Code:	**AI211**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of basic nature of data
2 To Extract the useful information from Data
3 Understanding of represent of data into meaningful information
4 Understanding some basics of statistics
5 How to handle tolerance in the Data
COURSE OUTCOMES
At the end of the course the students should be able to:
1 understand and critically assess available data using machine learning methods
2 Learn the basic concepts and techniques of Data Science and discover trends in both structured and unstructured data
3 Understand the concepts of supervised and unsupervised Learning.
4 Analyse complex problems using advanced analytics tools.
5 use of large volume data by extracting useful information and patterns and provide predictive insights.

UNIT I: Introduction to R and RStudio

Background, Getting Started, History of R and S, installing R and RStudio, Basic data types in R, Functions for reading and writing data. Using R for calculations. Using R to calculate summary statistics on data. Using R to generate random numbers. Variable types in R. Numeric variables, strings and factors, Statistics with R Analysing Data: Summary Statistics, Correlation and Covariance, Principal Components Analysis, Factor Analysis, Bootstrap Resampling. Probability Distributions: Normal Distribution, Common Distribution-Type Arguments, Distribution Function Families. Statistical Tests for Continuous and Discrete Data, Power Tests: Experimental Design Example, t-Test Design, Proportion Test Design, ANOVA Test Design.

UNIT II: Data structures: vectors, matrices, lists and data frames

Programming in R, Control structures- if-else, for loops, while, break, Repeat, next, Functions, Symbol binding, Scoping rules, Dates and times, the core data structures vectors, matrices, arrays, lists and data frames. sub-setting vectors, slicing arrays and drilling down on lists. lapply functions,

UNIT III: Reading data into R from various data sources

Loop functions- lapply, apply, mapply, tapply, split, Basic tools, Using the tools, reading from flat files (plain text), reading from database connections and reading from web sources, join command

UNIT IV: Statistical modeling functions

Simulation and profiling, Simulation- Generating Random numbers, simulating a Linear model, Random sampling, R profiler, Line

UNIT V: Writing your own functions

R function syntax, passing of variables into the function, and argument handling, brute force approaches, function evaluation s, notation that allows arguments to be passed on to other functions, functions that themselves take other functions as arguments.

Textbooks

1. Hands-On Programming with R: Write Your Own Functions and Simulations 1st Edition, Kindle Edition
2. Reference Books 1.R for Everyone: Advanced Analytics and Graphics

Database Management Systems Lab			
Course Code	**AI281**	**Course Credit**	**02**
Course Category	**CC**	**Course(U/P)**	**U**
Course year	**2U**	**Course Semester(U/P)**	**3U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**10+00**	**End Term Exam Hours:**	**03**
COURSE OBJECTIVES			
1.To explain basic database concepts, applications, data models, schemas and instances.			
2.To demonstrate the use of constraints and relational algebra operations. IV. Describe the basics of SQL and construct queries using SQL.			
3.To emphasize the importance of normalization in databases.			
4.To facilitate students in Database design			
5.To familiarize issues of concurrency control and transaction management.			
Course Outcomes			
At the end of the course the student should be able to understand the :			
1.Apply the basic concepts of Database Systems and Applications			
2. Use the basics of SQL and construct queries using SQL in database creation and interaction.			
3. Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system.			
4. Analyze and Select storage and recovery techniques of database system.			
5.Analyze of Transactions of DBMS			

List of Experiments

1. Introduction to MySQL, Postgre SQL, Microsoft SQL software.
2. Exercise on DML and DDL commands and create the following tables e.g, College records of 2 students, attendance records of a class.
3. Exercise on types of data constraints, using Postgre SQL/ MySQL.
4. Exercise on Group by and Order by clause using Postgre SQL/MySQL.
5. Exercise on implementing different functions, Arithmetic, Math and String.
6. Exercise on joins using normalization using single and multiple tables.
7. Exercise on views.
8. Exercise on procedure.
9. Exercise on triggers.
10. Case study using real life database applications.

IMAGE PROCESSING AND COMPUTER VISION LAB			
Course Code:	**AI283**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

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 design, 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.

List of Experiments:

1. Introduces the basic tools for working with images and the central Python modules.
2. Methods for detecting interest points in images and how to use them to find corresponding points and regions between images.
3. Transformations between images and methods for com- putting them. Examples range from image warping to creating panoramas.
4. Introduces how to model cameras, generate image projections from 3D space to image features and estimate the camera viewpoint.
5. How to compute 3D reconstructions from images.
6. Clustering methods and shows how to use them for grouping and organizing images based on similarity or content.
7. Shows how to build efficient image retrieval techniques that can store image representations and search for images based on their visual content.
8. Classifying image content and how to use them recognizing objects in images.

9. Techniques for dividing an image into meaningful regions using clustering, user interactions or image models.
10. Shows how to use the Python interface for the commonly used OpenCV computer vision library and how to work with video and camera input

R PROGRAMMING LAB			
Course Code:	**AI285**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**P**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 To perform the basic mathematical operations in r programming
2 Implementation of vector and List data objects operations.
3 Implementation of various operations on matrix, array, and factors in R.
4 Data Manipulation with dplyr package.
5 Study and implementation of Data Visualization with ggplot2.

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 design, 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.

List of Experiments

1. WAP to read data from various sources in a data frame.
2. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91
3. Write a R program to find the factors of a given number.
4. Write a R program to find the maximum and the minimum value of a given vector.
5. Write a R program to create a 5 x 4 matrix, 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.
6. Write a R program to get the statistical summary and nature of the data of a given data frame.
7. Write a R program to create inner, outer, left, right join(merge) from given two data frames.
8. Write a R program to save the information of a data frame in a file and display the information of the file
9. Build a simple web app using Shiny.
10. Write a R program to create a list containing a vector, a matrix and a list and give names to the 10 elements in the list. Access the first and second element of the list.
11. Load the data set and create a dashboard using shiny.
12. Connect the R script to dummy database and retrieve data from it and save it in a data frame.

SEMESTER IV

MACHINE LEARNING			
Course Code:	**AI202**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 05**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To introduce students to the basic concepts and techniques of Machine Learning			
2 To develop skills of using recent machine learning software for solving practical problem			
3 To gain experience of doing independent study and research.			
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:			
1Develop an appreciation for what is involved in Learning models from data			
2 Understand a wide variety of learning algorithms			
3 Apply the algorithms to a real problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models			
4 Understand how to evaluate models generated from data			
5 Understanding of ML models			

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 and Reference 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)

[3] 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

[4] 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

OPERATING SYSTEM			
Course Code:	**AI204**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs./Week):	**03+ 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):30	**30+ 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Understanding how Operating System is Important for Computer System.			
2 To make aware of different types of Operating System and their services.			
3 To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system			
4 To know virtual memory concepts and secondary memory management			
5 Understanding of Security & protection in Operating System			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1Understands the different services provided by Operating System at different level			
2They learn real life applications of Operating System in every field.			
3 Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock.			
They will learn different memory management techniques like paging, segmentation and demand paging etc.			
5 Perform implementation of protection mechanisms in operating system			

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management, processor management, device management, information management functions.

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm, and deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY &STORAGE MANAGEMENT

Basic Memory Management: Definition, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, partition, Fragmentation, Compaction, Paging, Segmentation.

UNIT IV UNIX/LINUX OPERATING SYSTEM: Development Of Unix/Linux, Role & Function Of Kernel, System Calls, Elementary Linux command & Shell Programming, Directory Structure, System Administration, Case study: Linux, Android Operating System

UNIT V SECURITY & PROTECTION: Security Environment, Design Principles of Security, And User authentication, Protection Mechanism: Protection Domain, Access Control List

Studying and Analyzing different case studies and research work.

Text Books:

[1]. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.

[2]. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.

Reference Books:

[3]. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.

[4]. Stallings William, Operating Systems, PHI, New Delhi, 1997.

[5]. S. Tanenbaum Modern Operating Systems, Pearson Education, 3rd edition, 2007.

[6]. Nutt, Operating System, Pearson Education, 2009.

[7]. S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 2nd edition, 2007.

Computer Interfacing and Embedded Systems			
Course Code:	**AI206**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

1. **COURSE OBJECTIVES**
2. Analyze the asymptotic performance of algorithms.
3. Write rigorous correctness proofs for algorithms.
4. Demonstrate a familiarity with major algorithms and data structures.
5. Apply important algorithmic design paradigms and methods of analysis.
6. Synthesize efficient algorithms in common engineering design situations.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Argue the correctness of algorithms using inductive proofs and invariant
2. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
5. Analyze worst-case running times of algorithms using asymptotic analysis.

Unit - I

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

Unit - II

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

Unit - III

Programming Embedded Systems : Structure of embedded program, infinite loop, compiling, linking and locating, downloading and debugging.

Unit - IV

Embedded hardware : Memory map, i/o map, interrupt map, processor family, external peripherals, memory - RAM, ROM, types of RAM and ROM, memory testing, CRC, Flash memory.

Unit - V

Peripherals : Control and Status Registers, Device Driver, Timer Driver-Watchdog Timers, Embedded Operating System, Real-Time Characteristics, Selection Process.

Design and Development : Embedded System development environment - IDE, Types of file generated on cross compilation, disassembler / decompiler, simulator, emulator and debugging, embedded product development life-cycle, trends in embedded industry.

Books :

Programming Embedded Systems in C and C++, First Edition January, Michael Barr, O' Reilly Introduction to embedded systems, Shibu K V Tata McGraw-Hill. 2

References : Embedded Systems, Rajkamal, TataMcGraw-Hill

DESIGN & ANALYSIS OF ALGORITHMS			
Course Code:	**AI208**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Argue the correctness of algorithms using inductive proofs and invariant
2. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
5. Analyze worst-case running times of algorithms using asymptotic analysis.

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

QUANTUM COMPUTING			
Course Code:	**AI210**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs/Week):	**03+ 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):30	**45+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To know the management of various storage technologies
2. This course focuses on evolution of storage and implementation models, Storage devices principles.
3. To know the Storage classes (SAN, NAS. CAS) and Backup
4. To explore the techniques Need of virtualization.
5. To Memory virtualization - Storage virtualization – Data virtualization – Network virtualization
COURSE OUTCOMES
At the end of the course the students should be able to:
1 To explain the design of a data center and storage requirements
2. To discuss the various types of storage and their properties
3. To explain physical and virtualization of storage
4. To explain the backup, archiving with regard to recovery and business continuity.

Unit I

Introduction: Elementary quantum mechanics:, linear algebra for quantum mechanics, Quantum states in Hilbert space, The Bloch sphere, Density operators, generalized measurements, no-cloning theorem.

Unit II

Quantum correlations: Bell inequalities and entanglement, Schmidt decomposition, superdense coding, teleportation.

Unit III

Quantum cryptography: quantum key distribution

Unit IV

Quantum gates and algorithms: Universal set of gates, quantum circuits, Solovay-Kitaev theorem, Deutsch-Jozsa algorithm, factoring

Unit V

Programming a quantum computer:The IBMQ, coding a quantum computer using a simulator to carry out basic quantum measurement and state analysis.

Text-books

(1) Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University press, 2007.

(1) Chris Bernhardt, Quantum Computing for Everyone, The MIT Press,Cambridge, 2020

(2)David McMahon-Quantum Computing Explained-Wiley-Interscience , IEEE Computer Society (2008)

References

(1) Quantum Computation and Quantum Information, M. A. Nielsen &I.Chuang, Cambridge University Press (2013).

(2) Quantum Computing, A Gentle Introduction , Eleanor G. Rieffel and Wolfgang H. Polak MIT press (2014)

COMPUTER NETWORKS			
Course Code:	**AI212**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Describe how computer networks are organized with the concept of layered approach.
2. Implement a simple LAN with hubs, bridges and switches.
3. Analyze the contents in a given Data Link layer packet, based on the layer concept.
4. Describe what classless addressing scheme is.
5. Describe how routing protocols work.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
2. Have a basic knowledge of the use of cryptography and network security.
3. Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols.
4. Analyse, specify and design the topological and routing strategies for an IP based networking infrastructure
5. Have a working knowledge of datagram and internet socket programming

UNIT I INTRODUCTION AND PHYSICAL LAYER

Key concepts of computer network, transmission media, network devices, network topology, topology design issues, types of network: LAN, MAN, WAN, PAN, ISDN systems and ATM network, OSI-reference model, open system standards, characteristics of network, TCP/IP model, protocols and standards, encoding technique.

UNIT II SWITCHING AND DATA LINK LAYER

Circuit switching, packet switching, message switching, hybrid switching, and ATM switching, multiplexing techniques: TDMA, FDMA, WDMA, CDMA, data link layer: LLC &MAC level protocols and design issues, issues IEEE 802 LAN Standards, framing, CRC, error control, flow control, HDLC, ALOHA and performance issues. Frames relay networks and performance parameters.

UNIT III NETWORK LAYER

Network layer design issues, overview of IPv4 and IPv6, addressing: class full and classless, static and dynamic, subnet and super net, auto configuration through DHCP, routing

protocols: RIP, DVR,LSR, OSFP, BGP, congestion control algorithm, subnet concept, virtual LAN, ICMP, multicasting, mobile IP.

UNIT IV TRANSPORT LAYER

Port addressing schemes, connectionless and connection oriented services: TCP and UDP, wireless TCP, Congestion control, queue management, NAT, PAT, socket format at transport level, socket interface and programming.

UNIT V APPLICATION LAYER

Client server architecture, domain name services, application services: HTTP, TELNET, RLOGIN, FTP, CBR, NFS, SMTP, POP, IMAP, MIME, voice and video over IP, social issues- privacy, freedom of speech, copy right.

Text Books:

1. S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2008
2. Forouzan, B.A., Data Communication and Networking, Tata McGraw-Hill.
3. W. Stallings, Data and Computer Communications, 8th edition, Prentice Hall, 2007
4. Douglus E. ComerTCP/IP Principles, Protocols and Architecture, Pearson Education

MACHINE LEARNING USING PYTHON LAB			
Course Code	**AI282**	**Course Credit**	**02**
Course Category	**CC**	**Course(U/P)**	**U**
Course Year(U/P)	**2U**	**Course Semester(U/P):**	**4U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**10+00**	**End Term Exam Hours:**	**03**
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 student should be able to understand the :			
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. **I**nstallation 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

Operating System Lab			
Course Code:	**AI284**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs./Week):3	**03+ 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures(L + T):10	**10+ 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To learn about file management and different types of permission setup			
2 To understand how system processes work and how to manage them			
3 To learn & implement different Operating system algorithm			
4 Apply concept of Deadlock and its prevention.			
5 Apply concept of OS to develop Producer Consumer problem & real scenario problems			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Able to implement and analyses the performance of different algorithm of Operating Systems like CPU scheduling algorithm, page replacement algorithms, deadlock avoidance, detection algorithm and so on.			
2 Able to design and develop a course project that can have positive impact on environment or society or mankind.			
3 Demonstrate the various operations of file system.			
4 Apply the various methods in memory allocation and page replacement algorithm.			
5 Apply the process synchronous concept using message queue, shared memory, semaphore and Dekker's algorithm for the given situation			

LIST OF PROGRAMS

1. Program for file handling.
2. Program for Dining Philosophers Problem.
3. Program for Producer – Consumer Problem concept.
4. Program for First Come First Serve Algorithm.
5. Program for Shortest Job First Scheduling Algorithm
6. Program for Round Robin Scheduling Method.
7. Program for Priority Scheduling Algorithm.
8. Implement the concept of Fragmentation and Defragmentation.
9. Simulate Bankers Algorithm for Dead Lock Avoidance
10. Simulate Bankers Algorithm for Dead Lock Prevention

DESIGN AND ANALYSIS OF ALGORITHMS LAB			
Course Code:	**AI286**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Labs (Hrs/Week):	**2(3hrs)**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. Write sorting programs using Divide-and-Conquer techniques.
2. Implement to find the minimum cost spanning tree and shortest path using different Greedy techniques
3. Construct DFS, BFS programs and topological ordering using Decrease-and-Conquer technique
4. Implement knapsack, travelling salesperson
5. Design different searching & sorting techniques and finding the complexities.

COURSE OUTCOMES

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

1. Demonstrate Quick sort and Merge sort and calculate the time required to sort the elements.
2. Implement the topological ordering of vertices, travelling salesman problem and Knapsack problem
3. Construct programs to check graph is connected or not using BFS and DFS methods
4. Implement programs on divide and conquer, decrease and conquer
5. Experiment finding the minimum cost of spanning tree using Prim's algorithms and shortest path using Dijkstra' algorithm

PRACTICALS

(Note: Use any programming tools like C/Java/Python to execute.)

1. Sort a given set of elements using the Quick sort method and also analyseit's runtime complexity for different inputs.

2. Sort a given set of elements using merge sort method and also analyseit's runtime complexity for different inputs.

3. Write a program to obtain the topological ordering of vertices in a given digraph.

4. Implement travelling salesman problem and knapsack problem (0/1).

5. Print all the nodes reachable from a given starting node in a digraph using BFS method.

6.Check whether a given graph is connected or not using DFS method.

7.Write a program to implement binary search using divide and conquer technique

8.Write a program to implement insertion sort using decrease and conquer technique

9. Find minimum cost spanning tree of a given undirected path using a Prim's algorithm.

10. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

SEMESTER V

BIG DATA ANALYTICS			
Course Code:	**AI301**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester(U /P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Understand the Big Data Platform and its Use cases
2 Provide an overview of Apache Hadoop
3 Provide HDFS Concepts and Interfacing with HDFS
4 Provide hands on Hadoop Eco System
5 Apply analytics on Structured, Unstructured Data.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Identify Big Data and its Business Implications.
2 List the components of Hadoop and Hadoop Eco-System
3 Access and Process Data on Distributed File System
4 Manage Job Execution in Hadoop Environment
5 Develop Big Data Solutions using Hadoop Eco System

UNIT I: UNDERSTANDING BIG DATA:

Introduction to big data, application of big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT II: NOSQL DATA MANAGEMENT:

Introduction to NoSQL, aggregate data models, aggregates, key value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharding, master slave replication, peer-peer replication, sharding and replication, consistency, relaxing consistency, version stamps, MapReduce, partitioning and combining, composing MapReduce calculations.

UNIT III: BASICS OF HADOOP:

Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop distributed file system (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization, Avro file-based data structure

UNIT IV: MAP REDUCE APPLICATIONS:

Map Reduce workflows, unit tests with MRUnit , test data and local tests – anatomy of

Map Reduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, MapReduce types, input formats, output formats.

UNIT V: HADOOP RELATED TOOLS:

Hbase, data model and implementations, Hbase clients, Hbase examples – praxis. Cassandra, cassandra data model, cassandra examples, cassandra clients, Hadoop integration. Pig, Grunt, pig data model, Pig Latin, developing and testing Pig Latin scripts. Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation – HiveQL queries

Text Books:

[1]. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.

[2]. Big-Data Black Book, DT Editorial Services,
Wily India Reference Books:
[3]. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.

[4]. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012. 5. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.

[5]. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
7. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.

[6]. Eben Hewitt, "Cassandra: The Definitive Guide", O'Reilley, 2010. [7]. Alan Gates, "Programming Pig", O'Reilley, 2011

COMPILER DESIGN			
Course Code:	**AI303**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.
2 Understand relations between computer architecture and how its understanding is useful in design of a compiler.
3 How to construct efficient algorithms for compilers.
4 Provide an understanding of the fundamental principles in compiler design.
5 Learn the process of translating a modern high-level-language to executable code required for compiler construction.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.
2 Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.
3 Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.
4 Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.
5 Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.

UNIT I INTRODUCTION TO COMPILER:

Introduction to Compiler: Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical-analyzer generator, LEX compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, Capabilities of CFG.

UNIT II BASIC PARSING TECHNIQUES:

Basic Parsing Techniques**:** Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III SYNTAX-DIRECTED TRANSLATION:

Syntax-directed Translation:Syntax-directed Translation schemes, Implementation of Syntax- directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array references in arithmetic expressions, procedures call, declarations and case statements.

UNIT IV SYMBOL TABLES:

Symbol Tables: Data structure for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack allocation scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.

UNIT V CODE GENERATION:

Code Generation: Design Issues, the Target Language, Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator, Code optimization: Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.

Text books:

1. K. Muneeswaran, CompilerDesign,FirstEdition,Oxford University Press.
2. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, Tata McGraw-Hill, 2003.
3. Henk Alblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
4. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools”, Pearson Education
5. V Raghvan, “ Principles of Compiler Design”, TM

DEEP LEARNING AND REINFORCEMENT LEARNING			
Course Code:	**AI305**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
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 design, 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 INTRODUCTION

History, capabilities, the perceptron, Neural network learning: Back-Propagation, Practical network training, Auto encoders, Batch-normalization, Overfitting and generalization

UNIT II INTRODUCTION TO TENSORFLOW AND ACTIVATION FUNCTIONS

Computational Graph, Key highlights, Creating a Graph, Regression example, Gradient Descent, TensorBoard, Modularity, Sharing Variables, KerasPerceptrons: What is a Perceptron, XOR Gate, Sigmoid, ReLU, Hyperbolic Fns, Softmax Artificial Neural Networks: Introduction, Perceptron Training Rule, Gradient Descent Rule.

UNIT III GRADIENT DESCENT AND BACKPROPAGATION

Gradient Descent, Stochastic Gradient Descent, Backpropagation, Some problems in ANN Optimization and Regularization: Overfitting and Capacity, Cross-Validation, Feature Selection, Regularization, Hyperparameters.

UNIT IV INTRODUCTION TO CONVOLUTIONAL NEURAL NETWORKS

Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters, CNN applications Introduction to Recurrent Neural Networks: Introduction to RNNs, Unfolded RNNs, Seq2Seq RNNs, LSTM, RNN applications.

UNIT V ADVANCED DEEP ARCHITECTURES

Recurrent Neural networks (RNNs), Advanced RNN: LSTM, GRU, Generative Adversarial Networks (GANs), Advanced GANs Reinforcement Learning: Definition., Types and Algorithms.

Text Books:

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.
3. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
   Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.

OPERATIONAL INFORMATION SECURITY MANAGEMENT AND BIOMETRICS			
Course Code:	**AI307**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Knowledge of Code of Ethics and understanding security concepts.			
2 Understanding different security measures.			
3 Define an information security strategy and architecture.			
4 Understanding the concepts of Biometrics.			
5 understanding of biometric equipment and standards applied to security..			
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 Analyse and evaluate the security measures needs of an organization..			
3 Design and develop a security architecture for an organization.			
4 Understand the technological uplifts with biometrics compared to traditional securing Mechanisms..			
5 Gain knowledge in building blocks of research fields like Face ,Signature, Fingerprint, Ear, Iris.			

UNIT I OPERATIONAL SECURITYCONCEPTS

Code of Ethics, Organizational Code of Ethics, Types of Operations: Managerial, Technical, and Operational, Deterrent, Preventative, Detective, and Corrective Controls. Code of ethics applies to security practitioners, applying ethical principles, understand security concepts: Breach, Confidentiality, Integrity, availability, Non-Repudiation, Privacy.

UNIT II SECURITY MEASURES

Security, Password and account management, Cryptography technology, digital and Malware (computer virus, spyware, bot, worm, malicious adware, crack tool, etc.) countermeasures, Application security measures, Database security measures, Network security measures, System security measures, Physical security measures, Access control, Attack techniques

UNIT III SECURITY ARCHITECTURE

Designing of the security architecture, Operations management, Security technologies, Network protocols, topology, routing, and network hardware, Software operation required by operators, External security diagnostic services ,Security incidents and accidents ,Network attacks , Log management ,System log and access log .\

UNIT IV BIOMETRICS

Introduction of Biometric traits and its aim, image processing basics, basic image operations, filtering, enhancement, sharpening, edge detection, smoothening, enhancement, thresholding, localization. Fourier series, DFT, inverse of DFT. Biometric system, identification and verification. FAR/FRR, system design issues. Positive/negative identification. Biometric system security, authentication protocols, matching score distribution, ROC curve, DET curve, FAR/FRR curve. Expected overall error, EER, biometric myths and misrepresentations.

UNIT V BIOMETRIC SECURITY MANAGEMENT

Biometric system security, Biometric system vulnerabilities, circumvention, covert acquisition, quality control, template generation, interoperability, data storage. Recognition systems: Face, Signature, Fingerprint, Ear, Iris

REFERENCE BOOKS:

1. Smith,“ Elementary Information Security”, (2011, Jones & Bartlett learning).
2. Mark Dowd, John McDonald, Justin Schuh,„“ The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities”, 2007, Addison-Wesley
3. Paul van Oorschot, “Computer Security and the Internet: Tools and Jewels ‘’(2020, Springer).
4. Stallings and Brown, Computer Security: Principles and Practice, 3/e (2014, Prentice Hall).
5. Mark Dowd, John McDonald, Justin Schuh,„“ Biometrics: Theory, Methods, and Applications, Nikolaos V. Boulgouris, Konstantinos N. Plataniotis, Evangelia Micheli-Tzanakou

ELECTIVE 1

COMPUTER GRAPHICS			
Course Code:	**AI309**	**Course Credits:**	**3**
Course Category:	**E1/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics. c.
2. Extract scene with different clipping methods and its transformation to graphics display device.
3. Use of geometric transformations on graphics objects and their application in composite form.
4. Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Explain the core concepts of computer graphics, including viewing, projection, perspective, modelling and transformation in two and three dimensions.
2. Apply the concepts of colour models, lighting and shading models, textures, ray tracing, hidden surface elimination, anti-aliasing, and rendering.
3. Interpret the mathematical foundation of the concepts of computer graphics.
4. Describe the fundamentals of animation, parametric curves and surfaces, and spotlighting.
5. Identify a typical graphics pipeline and apply graphics programming techniques to design and create computer graphics

UNIT 1 BASICS OF COMPUTER GRAPHICS

Introduction, What is computer Graphics?, Area of Computer Graphics, Design and Drawing, Animation Multimedia applications, Simulation, How are pictures actually stored and displayed, Difficulties for displaying pictures.

UNIT 2 GRAPHIC DEVICES

Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube, Tablets, The light Pen, Three Dimensional Devices C Graphics Basics Graphics programming, initializing the graphics, C Graphical functions, simple programs SIMPLE LINE DRAWING METHODS Point Plotting Techniques, Qualities of good line drawing algorithms, The Digital Differential Analyzer (DDA), Bresenham's Algorithm, Generation of Circles

UNIT 3 TWO DIMENSIONAL TRANSFORMATIONS

What is transformation?, Matrix representation of points, Basic transformation, Need for Clipping and Windowing, Line Clipping Algorithms, The midpoint subdivision Method, Other Clipping Methods, Sutherland - Hodgeman Algorithm, Viewing Transformations

UNIT 4 GRAPHICAL INPUT TECHNIQUES

Graphical Input Techniques, Positioning Techniques, Positional Constraints, Rubber band Techniques THREE DIMENSIONAL GRAPHICS Need for 3-Dimensional Imaging, Techniques for 3-Dimesional displaying, Parallel Projections, Perspective projection, Intensity cues, Stereoscope effect, Kinetic depth effect, Solid Area Scan Conversion, Scan Conversion of Polygons, Algorithm Singularity, Three Dimensional transformation, Translations, Scaling, Rotation, Viewing Transformation, The Perspective, Algorithms, Three Dimensional Clipping, Perspective view of Cube

UNIT 5 COMPUTER GRAPHICS HIDDEN SURFACE REMOVAL

Need for hidden surface removal, The Depth - Buffer Algorithm, Properties that help in reducing efforts, Scan Line coherence algorithm, Span - Coherence algorithm, Area-Coherence Algorithms, Warnock's Algorithm, Priority Algorithms 2

Text Books:

1. Watt, Alan. 3D Computer Graphics. Addison-Wesley, 1999
2. Buss, Samuel R. 3D Computer Graphics: A Mathematical Introduction with OpenGL. 2003
3. Akenine-Moller, Tomas, Eric Haines and Naty Hoffman. Real-Time Rendering.
4. Computer Graphics, D.Hearn And P.Baker - Pearson Eduction - C Version
5. Computer Graphics, with OpenGL Hearn and Baker, - Pearson

INTRODUCTION TO BRAIN AND NEUROSCIENCE			
Course Code:	**AI311**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Knowledge of neuro imaging and special applications.			
2 Understanding of image visualization			
3 Knowledge of MRI data preparation.			
4 Study for different ways to design an an MRI experiment			
5 Understanding of PET imaging.			
COURSE OUTCOMES			
1 At the end of the course the students should be able to:			
2 Students will be able to have cursory knowledge of strengths and weaknesses of various brain imaging methods			
3 He/she will be able to account for the physical and physiological basis of fMRI measurements			
4 Be able to account for how fMRI data are prepared for analyses, and describe different ways to analyze fMRI data			
5 Be able to account for different ways to design an fMRI experiment in relation to various questions on brain function			
6 Have cursory knowledge of in what way structural brain imaging can inform questions on brain function			

UNIT I

What is Neuroscience, Brain Analogy, Introduction to Neurophysiology, Basic Operation of Human Brain Biomedical Imaging Techniques, Micro Electro-mechanical Systems, Basics of R programming, Installation of required packages, Structural Neuro anatomy of the Human Brain, Functional Neuro anatomy of the Human Brain, Principles and Methods of Neuro imaging, Experimental design, and special applications in neuro imaging.

UNIT II

Visualization of Images, Inhomogeneity Correction, Brain Extraction/Segmentation, Image Registration, Tissue-Class Segmentation, Intensity Normalization, Segmentation of MS Lesions, Image Harmonization, Starting with Raw (DICOM) Data.

UNIT III

Introduction to fMRI, Basic Physical Principles, Image Formation, Contrast Mechanisms and Pulse Sequences, From Neuronal to Hemodynamic Activity, BOLD fMRI, Introduction to FSL, Signal and Noise; Pre-processing, Statistics I: General Linear Model, Statistics II: Advanced Techniques, Advanced MR Models, Future of fMRI

UNIT IV

Introduction to EEG, assessing cognitive mechanisms via EEG-derived methods, the neural basis of the EEG signal and its dependent measures, Recording EEG: principles and techniques, The initial steps of EEG analysis, Spectral analysis and time-frequency analysis, Experimental design for EEG.

UNIT V

Introduction to PET-CT, Fundamentals of PET, Principles of PET, Recent advances of PET imaging in clinical radiation oncology, Procedure Guideline for Tumour Imaging with 18F-FDG PET/CT, Diagnosis and Analysis of brain through PET-CT.

Text Books:

1. The computational brain by P.S. Churchland and T.J. Sejnowski, MIT Press.
2. Theoretical Neuroscience by P. Dayan and L.F. Abbott, MIT Press
3. The Journal of Computational Neuroscience, Kluwer Academic Publishers
4. Neural Computations, MIT Press

STOCHASTIC PROCESS			
Course Code:	**AI313**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P)**	**U**
Course Year(U/P):	**U**	**Course Semester(U/P)**	**5U**
No. of Lectures +Tutorials(Hrs/Week):	**3+00**	**MidSem. Exam Hours**	**1.5**
Total No. of Lectures(L+ T):	**45+00**	**EndSem. Exam Hours**	**3**

COURSE OBJECTIVES
1. Classify Markov chains in dicrete and continuous time with respect to state diagrams, recurrence and trancience, states, periodicty and irreducibility
2. Conduct calculations with transition probabilities and transition intensities
3. Calculate absorbtion probabilities and the expected time to absorbtion for Markov chains.
4. Choose a proper Markov model and conduct proper calculations for different applications, especially regarding the modeling of birth-and-death processes
5. Apply the Markov chain Monte Carlo method and hidden Markov models.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Thoroughly describe the theory of stochastic processes, especially for Markov processes.
2. Define Markov chains in discrete and continuous time.
3. Define the existence and uniqueness of stationary and asymptotic distributions for Markov chains, and whenever applicable derive the distributions as solutions to balance equations.
4. Thoroughly explain the meaning of Markov processes with continous state space, especially for rownian motion and diffusion processes.
5. Critically describe the connection between the theory of Markov processes and differential equations

Unit-I

Intermediate Probability: Manipulating RVs Conditioning RVs, Conditional Distribution of a RV ,Computing probabilities and expectations by conditioning , Information Systems Application: Time-to-a-pattern for password security Inequalities and Limits of Events, RVs, Distributions ,Inequalities: Markov, Chebyshev, Jensen, Holder.

Unit II

Convergence of Sets, Probabilities and Distributions; Distributions of extreme, Marketing Application: Multinomial choice model Classifying and Ordering RVs, Increasing failure rate and Polya densities, Sochastic order, Hazard rate order, Likelihood ratio order, Convex order Risk Applications: Risk comparisons with convex order

Unit III:

Stochastic Processes: Indexing RVs Markov Chains, Markovian property and Transition probabilities, Irreducibility and Steady-State probabilities

Unit IV

Generic Applications: Hidden Markov Chains Exponential Distribution and Poisson Process , Construction of Poisson Process from Exponential Distribution , Thinning and Conditional

Arrival Times ,Service Applications: Waiting Times Normal Distribution and Brownian Process. Construction of Brownian Process from Normal Distribution, Hitting Times and Maximum Values, Finance Applications: Option Pricing and Arbitrage Theorem

Unit V

Introduction to stochastic differential equations, Itˆo calculus, Fokker-Planckequation, Ornstein-Uhlen beckprocess.

Texts Books

[1]M.Lefebvre.AppliedStochasticProcesses.Springer,2007

[2]Z.Brze´zniak,T.Zastawniak.BasicStochasticProcesses.Springer,1999

[3]E.Parzen.StochasticProceses.SIAM,1999

[4]R. Durrett.EssentialsofStochasticProcesses.Second ed., Springer, 2012

[5]S.Ross.IntroductiontoProbabilityModels.Eighthed.,Elsevier,2003

COMPUTER BASED NUMERICAL AND STATISTICAL TECHNIQUES			
Course Code:	**AI315**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course(U/P)**	**U**
Course Year(U/P):U	**3U**	**Course Semester(U/P):**	**5U**
No.of Lectures+ Tutorials(Hrs/Week):	**03+00**	**MidSem.ExamHours:**	**1.5**
Total No. of Lectures(L+T):30	**45+00**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1. Formulate and solve linear programming problems and operations with nonlinear expressions.			
2.Able to find the mean and the variance of a random variable			
3. Able to find the confidence interval for the mean of a normal population from a sample			
4. Ability to solve financial math problems			
5 Ability to solve basic problems in probability and statistics.			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1.Understandthesample regression line			
2.Learnto solve the equation by Newton Raphson Method			
3.Understand and Develop design and problem solving skills with application to statistical methods			
4. Familiarity with key computer based numerical for modeling and rendering data			
5.Performthe basic principles of implementing computer statistical primitives			

Unit-I

Introduction: Numbers and their accuracy, Computer Arithmetic, Mathematical preliminaries, Errors and their Computation, General error formula, Error in a series approximation Solution of Algebraic and Transcendental Equation: Bisection Method, Iteration method, Method of false position, Newton-Raphson method,Methods of finding complex roots, Muller's method, Rate of convergence of Iterativemethods,Polynomial Equations.

Unit-II

Interpolation: Finite Differences, Difference tables, Polynomial Interpolation: Newton's forward and backward formula, Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation.

Unit-III

Numerical Integration and Differentiation: Introduction, Numerical differentiation Numerical Integration: Trapezoidalrule, Simpson's1/3 and 3/8rule, Boole'srule, Waddle'srule.

Unit-IV

Solution of differential Equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods, Predictor Corrector Methods, Automatic Error Monitoring and Stability of solution

Unit-V

Statistical Computation: Frequency chart, Curve fitting by method of least squares, fitting of straight lines, polynomials ,exponential curve setc, Data fitting with Cubic splines, Regression Analysis, Linear and Non linear Regression, Multiple regression, Statistical Quality Control methods.

References:

1. RajaramanV, "Computer Oriented Numerical Methods", Pearson Education
2. Gerald & Whealey, "Applied Numerical Analyses",AW
3. Jain, Iyengarand Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
4. GrewalBS, "Numerical methods in Engineering and Science" ,Khanna Publishers, Delhi
5. T Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods, TMH
6. Pradip Niyogi, "Numerical Analysis and Algorithms",TMH
7. Francis Scheld,"Numerical Analysis",TMH
8. SastryS.S,"Introductory Methods of Numerical Analysis",Pearson Education.
9. GuptaC.B.,Vijay Gupta,"Introduction to Statistical Methods",Vikas Publishing.
10. Goyal, M, "Computer Based Numerical and Statistical Techniques", Firewall Media, NewDelhi.

SEQUENCE MODELS			
Course Code:	**AI317**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P)**	**U**
Course Year(U/P):	**U**	**Course Semester(U/P)**	**6U**
No. of Lectures +Tutorials(Hrs/Week):	**3**	**MidSem. Exam Hours**	**1.5**
Total No. of Lectures(L+ T):	**45**	**EndSem. Exam Hours**	**3**
COURSE OBJECTIVES			
1. To understand the basics of language processing			
2. To learn about language models, sequence labelling tasks			
3. To learn about parsing, machine translation systems4			
4. To learn about Q/A systems, Summarization, Chatbots			
5. To learn of summarization of model working.			
COURSE OUTCOMES			
Students will learning the following:			
1. How a language model works			
2. How a POS tagging system, NER systems works			
3. How to design a Parser for a given language			
4. How to design a statistical as well as neural machine translation model			
5. How a Q/A system and Chatbot works			

UNIT 1

Define sequence data, Applications of Sequence Models: Speech recognition, Sentiment Classification, Video Activity Recognition, Recurrent Neural Networks (RNNs), RNN Architectures, Sequence Modeling Problems, Sequence Classification.

UNIT 2

Modeling Sequence Learning Problems, Recurrent Neural Networks, practical applications of sequence modeling, Auto-Completion, Parts of Speech Tagging, Modeling Sequence Learning Problems.

UNIT 3

Sequence Models & Attention Mechanism, Sequence datap: Image Captioning, Time Series, Natural Language Processing, Machine Translation, Speech recognition, Music generation ,DNA sequence analysis

UNIT 4

Recurrent Neural network: Graphical Representation of RNN, Unfold Representation, Weight Matrices parameters in RNN Sequence Modeling Tasks, Different types of Sequence Modeling Tasks, How to train a RNN, Back propagation through time, Vanishing and Exploding gradient

UNIT 5

Long Short Term Memory (LSTM) Main ideas, Graphical Representation of LSTM, Gated Recurrent Units (GRU), Auto-encoders for language translation, Cho's model, Sutskever's model, Improvement with Attention layer

Text Books:

1. Sequence Learning: Paradigms, Algorithms, and Applications: 1828, by Ron Sun (Editor), C.Lee Giles (Editor)
2. Sequence to Sequence Learning with Neural Networks, Ilya Sutskever, Oriol Vinyals, Quoc V. Le
3. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio

ELECTIVE 2

Bayesian Data Theory			
Course Code:	**AI 319**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. An understanding of selected advanced topics in Bayesian statistics;
2. Preparation for a research or industry career in statistics and data science;
3. Developed the mathematical and computational skills needed for further research or applied work in statistics and data science;
4. Familiarity with several major texts in Bayesian statistics

COURSE OUTCOMES

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

1. In addition to learning specific skills that will assist students in their future careers in science
2. These include: - problem-solving skills: the ability to engage with unfamiliar problems and identify relevant solution strategies
3. analytical skills: the ability to construct and express logical arguments and to work in abstract or general terms to increase the clarity and efficiency of analysis
4. Bayesian statistical learning including credible regions, prior choice, comparisons of means and proportions, multi-model inference and model selection.

UNIT-1 :A Brief Tour of Bayesian Inference and Decision Theory

Describe the elements of a decision model ,Refresh knowledge of probability, Apply Bayes rule for simple inference problems and interpret the results, Bayesians believe inference cannot be separated from decision-making ,Compare Bayesian and frequentist philosophies of statistical inference , Compute and interpret the expected value of information (VOI) for a

decision problem with an option to collect information , Download, install and use R statistical software

Unit 2: Random Variables, Parametric Models, and Inference from Observation

Event, Random variable (univariate and multivariate), Joint, conditional and marginal distributions, Independence; conditional independenc, Mass and density functions; cumulative distribution functions, Measures of central tendency and spread, Use a graph to show conditional dependence and independence of random variables, Be familiar with common parametric statistical models, Continuous distributions: Normal, Gamma, Exponential, Uniform, Beta, Dirichlet, Discrete distributions: Bernoulli, Binomial, Multinomial, Poisson, Negative Binomial

Unit 3: Statistical Models with a Single Parameter

Define a conjugate pair of distributions,Canonical Statistical Inference Problem,Bayesian Approach to Canonical Inference Problem,Bayes Rule for Continuous Distributions,Poisson Distribution ,Gamma Distribution,Parameterizations for Gamma Distribution ,Posterior Distribution for Λ

Unit 4: Monte Carlo Approximation

Describe the Monte Carlo method and explain why it works, Estimate common aspects of a distribution by sampling from the distribution , Expected value, Variance and standard deviation, Quantiles and credible intervals , Other interesting quantities, Use Monte Carlo to predict future observations conditional on past observations ,Apply posterior predictive sampling to evaluate adequacy of a model and investigate departures from the model

Unit 5:

The Normal Model, Markov Chain Monte Carlo , Hierarchical Bayesian Models ,Bayesian Regression and Analysis of Variance, Multinomial Distribution and Latent Groups Hypothesis Tests, Bayes Factors, and Bayesian Model Averaging

Text Book :

1. Bayesian Data Analysis, by Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin.
2. Statistical Rethinking A Bayesian Course with Examples in R and STAN, Richard McElreath
3. Bayesian Statistics the Fun Way,Understanding Statistics and Probability with Star Wars, LEGO, and Rubber Ducks,Will Kurt
4. Bayesian Analysis,A Practical Introduction

SPEECH ANALYSIS AND SYSTEMS			
Course Code:	**AI 321**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course(U/P)**	**U**
Course Year(U/P):U	**3U**	**Course Semester(U/P):**	**5U**
No.of Lectures+ Tutorials(Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5**
Total No.of Lectures(L+T):30	**45+00**	**End Sem. Exam Hours:**	**3**

COURSEOBJECTIVES
1. To understand the fundamentals of the speech processing
2. Explore the various speech models
3. Gather knowledge about the phonetics and pronunciation processing
4. Perform wavelet analysis of speech
5. To understand the concepts of speech recognition

COURSEOUTCOMES
At the end of the course the students should be able to:
1. Create new algorithms with speech processing
2. Derive new speech models
3. Perform various language phonetic analysis
4. Create a new speech identification system
5. Generate a new speech recognition system

UNIT 1 Introduction

Introduction - knowledge in speech and language processing - ambiguity - models and algorithms - language - thought - understanding - regular expression and automata - words & transducers – N grams

UNIT 2Speech Modelling

Word classes and part of speech tagging – hidden markov model – computing likelihood: the forward algorithm – training hidden markov model – maximum entropy model – transformation-based tagging – evaluation and error analysis – issues in part of speech tagging – noisy channel model for spelling

UNIT 3Speech Pronunciation and Signal Processing

Phonetics - speech sounds and phonetic transcription - articulatory phonetics - phonological categories and pronunciation variation - acoustic phonetics and signals - phonetic resources - articulatory and gestural phonology

UNIT 4Speech Identification

Speech synthesis - text normalization - phonetic analysis - prosodic analysis – diphone waveform synthesis - unit selection waveform synthesis - evaluation

UNIT 5Speech Recognition

Automatic speech recognition - architecture - applying hidden markov model - feature extraction: mfcc vectors - computing acoustic likelihoods - search and decoding - embedded training - multipass decoding: n-best lists and lattices- a* (_stack‘) decoding - context-dependent acoustic models: triphones - discriminative training - speech recognition by humans

TEXT BOOK:

1. Daniel Jurafsky and James H. Martin, ― Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition‖, Person education,2013.

REFERENCES

1. Kai-Fu Lee, ―Automatic Speech Recognition‖, The Springer International Series in Engineering and Computer Science, 1999.
2. Himanshu Chaurasiya, ―Soft Computing Implementation of Automatic Speech Recognition‖, LAP Lambert Academic Publishing, 2010.
3. Claudio Becchetti, KlucioPrina Ricotti, ―Speech Recognition: Theory and C++ implementation‖,Wiley publications 2008.
4. Ikrami Eldirawy ,Wesam Ashour, ―Visual Speech Recognition‖, Wiley publications , 2011

Augmented and Virtual Reality			
Course Code:	**AI322**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P):**	**U**
Course Year(U/P):	**3U**	**Course Semester(U/P):**	**6U**
No. of Lectures +Tutorials (Hrs/Week):	**03+00**	**MidSem. ExamHours:**	**1.5**
Total No .of Lectures(L+T):	**45+00**	**EndSem. ExamHours:**	**3**

COURSE OBJECTIVES
1.Learn the fundamental Computer Vision, Computer Graphics and Human-Computer interaction Techniques related to VR/AR
2.Review the Geometric Modeling Techniques
3.Review the Virtual Environment
4.Use of various types of Hardware and Software in Virtual Reality systems
5.Simulate and Apply Virtual/Augmented Reality to varieties of Applications

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand fundamental Computer Vision, Computer Graphics and Human Computer Interaction Techniques related to VR/AR
2. Understand Geometric Modeling Techniques.
3. Understand the Virtual Environment
4. Analyze and evaluate VR/AR Technologies
5. Apply various types of Hardware and Software in Virtual Reality systems and design and formulate Virtual/Augmented Reality Applications

Unit-I

Introduction to Virtual Reality (VR) Virtual Reality and Virtual Environment, Computer graphics, Real time computer graphics, Flight Simulation, Virtual environment requirement, benefits of virtual reality, Historical development of VR, Scientific Landmark

Unit II

Computer Graphics and Geometric Modelling: The Virtual world space, positioning the virtual observer, the perspective projection, human vision, stereo perspective projection, Color theory, Conversion From 2D to 3D, 3D space curves, 3D boundary representation, Simple 3D modelling, 3D clipping, Illumination models, Reflection models, Shading algorithms, Geometrical Transformations: Introduction, Frames of reference, Modelling transformations, Instances, Picking, Flying, Scaling the VE, Collision detection

Unit III:

Virtual Environment: Input/Output Devices: Input (Tracker, Sensor, Digital Gloves, Movement Capture, Videobased Input, 3D Menus & 3D Scanner, etc.), Output (Visual/Auditory/Haptic Devices) Generic VR system: Introduction, Virtual environment, Computer environment, VR technology, Model of interaction, VR Systems, Animating the Virtual Environment.

Unit IV

Augmented Reality (AR): Taxonomy, Technology and Features of Augmented Reality, AR Vs VR, Challenges with AR, AR systems and functionality, Augmented Reality Methods, Visualization Techniques for Augmented Reality, Enhancing interactivity in AR Environments, Evaluating ARsystems

Unit V

Development Tools and Frameworks: Human factors: Introduction, the eye, the ear, the somatic senses Hardware: Introduction, sensor hardware, Head-coupled displays, Acoustic hardware, Integrated VR systems Software: Introduction, Modelling virtual world, Physical simulation, VR toolkits, Introduction to VRML, AR / VR Applications: Introduction, Engineering, Entertainment, Science, Training, Game Development

TEXT BOOKS:

1. Coiffet, P., Burdea, G. C., (2003), "Virtual Reality Technology," Wiley-IEEE Press, ISBN: 9780471360896 2. Schmalstieg, D., Höllerer, T., (2016), "Augmented Reality: Principles & Practice," Pearson, ISBN: 9789332578494
2. Norman, K., Kirakowski, J., (2018), " Wiley Handbook of Human Computer Interaction," Wiley-Blackwell, ISBN: 9781118976135
3. LaViola Jr., J. J., Kruijff, E., McMahan, R. P., Bowman, D. A., Poupyrev, I., (2017), "3D User Interfaces: Theory and Practice," Pearson, ISBN: 9780134034324
4. Fowler, A., (2019), "Beginning iOS AR Game Development: Developing Augmented Reality Apps with Unity and C#," Apress, ISBN: 9781484246672
5. Hassanien, A. E., Gupta, D., Khanna, A., Slowik, A., (2022), "Virtual and Augmented Reality for Automobile Industry: Innovation Vision and Applications," Springer, ISBN: 9783030941017

GRAPH THEORY			
Course Code:	**AI323**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To understand the concept of vertex connectivity and edge connectivity in graphs.			
2 To develop the under-standing of Geometric duals in Planar Graphs			
3 To understand the concept of digraphs, Euler digraphs and Hamiltonian digraphs			
4 To understand the concept of matrices in graphs like Incidence matrix, Adjacency matrix, Cycle matrix, etc			
5 To introduce the idea of coloring in graphs.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Basic knowledge and understanding of the analysis and graphs.			
2 Understand and explain the basic concepts of graph theory.			
3 Apply the basic concepts of mathematical logic			
4 Analyze the basic concepts of mathematical logic.			
5 Evaluate some real time problems using concepts of graph theory.			

UNIT I

Graphs, Sub graphs, some basic properties, various example of graphs & their sub graphs, walks, path & circuits, connected graphs, disconnected graphs and component, euler graphs, various operation on graphs, Hamiltonian paths and circuits, the traveling sales man problem.

UNIT II

Trees and fundamental circuits, distance diameters, radius and pendent vertices, rooted and binary trees, on counting trees, spanning trees, fundamental circuits, finding all spanning trees of a graph and a weighted graph, algorithms of primes, Kruskal and Dijkstra Algorithms.

UNIT III

Cuts sets and cut vertices, some properties, all cut sets in a graph, fundamental circuits and cut sets connectivity and separability, network flows Planer graphs, combinatorial and geometric dual: Kuratowski graphs, detection of planarity, geometric dual, Discussion on criterion of planarity, thickness and crossings.

UNIT IV

Vector space of a graph and vectors, basis vector, cut set vector, circuit vector, circuit and cut set subspaces, Matrix representation of graph – Basic concepts; Incidence matrix, Circuit matrix, Path matrix, Cut-set matrix and Adjacency matrix.

UNIT V

Colouring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, matching, covering, four color problem Discussion of Graph theoretic algorithm wherever required.

Text Books:

1. Deo, N, Graph theory with applications to Engineering and Computer Science, PHI
2. Gary Chartrand aend Ping Zhang, Introduction to Graph Theory, TMH
3. Robin J. Wilson, Introduction to Graph Theory, Pearson Education
4. Harary, F, Graph Theory, Narosa
5. Bondy and Murthy: Graph theory and application. Addison Wesley.

DISTRIBUTED DATABASE			
Course Code:	**AI325**	**Course Credits:**	**33**
Course Category:CC	**CC**	**Course(U/P)**	
Course Year(U/P):U	**23U**	**Course Semester(U/P):**	**5U**
No.of Lectures+Tutorials Hrs/Week):	**03+00**	**MidSem. ExamHours:**	**11.5**
Total No.of Lectures(L+T):	**445+00**	**EndSem. ExamHours:**	**33**
COURSEOBJECTIVES			
1.Understand distributed database systems architecture and design			
2. able to apply methods and techniques for distributed quey processing and optimisation			
3. Understand the broad concepts of distributed transaction process			
4. Understand the basic concepts of Data warehousing and OLAP technology			
5. able to apply methods and techniques for association analysis, data classification and clustering			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Understand theoretical and practical aspects of distributed database systems.			
2. Study and identify various issues related to the development of distributed database system.			
3. Understand the design aspects of object-oriented database system and related development.			
4. Understand theoretical and practical aspects of distributed database systems.			
5. Knowledge and understanding of the content and techniques			

Unit-1 Introduction

Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas. Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDMBS Architecture. Distributed Database Design: Alternative Design Strategies, Distribution Design issues, Fragmentation, Allocation.

Unit-2 Query processing and decomposition

Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data. Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

Unit-3 Transaction Management

Transaction Management: Definition, properties of transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms & algorithms, time - stamped & optimistic concurrency control Algorithms, deadlock Management.

Unit-4 Distributed DBMS Reliability

Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning. Parallel Database Systems: Parallel database system

architectures, parallel data placement, parallel query processing, load balancing, database clusters.

Unit-5 Distributed object Database Management Systems

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing**.** Object Oriented Data Model**:** Inheritance, object identity, persistent programming languages, persistence of objects, comparison OODBMS and ORDBMS

TEXT BOOKS:

1. M. Tamer OZSU and PatuckValduriez: Principles of Distributed Database Systems, Pearson Edn. Asia, 2001.
2. Stefano Ceri and Giuseppe Pelagatti: Distributed Databases, McGraw Hill.

REFERENCE BOOKS:

1. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: "Database Systems: The Complete Book", Second Edition, Pearson International Edition

PATTERN RECOGNITION			
CourseCode:	**AI327**	**CourseCredits:**	**3**
CourseCategory:	**E1**	**Course(U/P)**	**U**
CourseYear(U/P):	**3U**	**CourseSemester(U/P):**	**5U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem.ExamHours:**	**1.5**
Total No.ofLectures (L+T):	**45+00**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1.To understand different pattern and mathematical foundation.			
2.Understand statistic approach.			
3.To understand estimation method.			
4.Understand KNN, Nearest neighbor rule and other non parametric technique.			
5.Understand different clustering algorithm along with unsupervised learning			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1.Mathematically approach different pattern recognition			
2.Apply statistical approach of pattern recognition.			
3.Perform different estimation methods			
4.Apply non parametric techniques like KNN fuzzy classification etc.			
5.Implement unsupervised learning and clustering			

UNIT I INTRODUCTION

Basics of pattern recognition, Design principles of pattern recognition system, Learning and adaptation, Pattern recognition approaches, Mathematical foundations–Linear algebra, Probability Theory, Expectation, mean and covariance, Normal distribution, multivariate normal densities, Chi squared test.

UNIT II STATISTICAL PATTEN RECOGNITION: Bayesian DecisionTheory, Classifiers, Normal density and discriminant functions.

UNIT III PARAMETER ESTIMATION METHODS

Maximum-Likelihood estimation, Bayesian Parameter Estimation, Dimension reduction methods– Principal Component Analysis (PCA), Fisher Linear discriminant analysis, Expectation-maximization(EM), Hidden Markov Models(HMM), Gaussian mixture models.

UNIT IV NON PARAMETRIC TECHNIQUES:

Density Estimation, Parzen Windows, K-Nearest Neighbor Estimation, Nearest Neighbor Rule, Fuzzy classification.

UNIT V UNSUPERVISED LEARNING & CLUSTERING:

Criterion functions for clustering, Clustering Techniques: Iterative square-error partitional clustering–Kmeans, agglomerative hierarchical clustering, Cluster validation.

REFERENCES:

1. Richard O.Duda, PeterE.Hart and David G.Stork,"Pattern Classification",2nd Edition, John Wiley, 2006.
2. C.M.Bishop,"Pattern Recognition and Machine Learning",Springer, 2009.
S.The odoridis and K. Koutroumbas, "Pattern Recognition", 4th Edition,Academic Press

BIG DATA ANALYTICS LAB			
Course Code	**AI381**	**Course Credit**	**02**
Course Category	**CC**	**Course(U/P)**	**U**
Course year(U/P)	**3U**	**Course Semester**	**5U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**10+00**	**End Term Exam Hours:**	**03**
COURSE OBJECTIVES			
1. To study the basic technologies that forms the foundations of Big Data.			
2. To study the programming aspects of cloud computing with a view to rapid prototyping of complex applications.			
3. To understand the specialized aspects of big data including big data application, and big data analytics.			
4. To study different types Case studies on the current research and applications of the Hadoop and big data in industry			
5. Understand the HIVE database and Tables			
COURSE OUTCOMES			
1.At the end of the course the student should be able to understand the :			
2.Student must be Able to understand the building blocks of Big Data			
3.Student must be able to articulate the programming aspects of cloud computing(map Reduce etc)			
4.Student must be able to understand the specialized aspects of big data with the help of different big data applications			
5. Student must be able to represent the analytical aspects of Big Data			

LIST OF EXPERIMENTS:

1. Installation of VMWare to setup the Hadoop environment and its ecosystems.
2. Perform setting up and Installing Hadoop in its three operating modes. i. Standalone. ii. Pseudo distributed. iii. Fully distributed.
3. Use web based tools to monitor your Hadoop setup.
4. Implementing the basic commands of LINUX Operating System – File/Directory creation, deletion, update operations
5. Implement the following file management tasks in Hadoop: i. Adding files and directories ii. Retrieving files iii. Deleting files Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.
6. Run a basic word count Map Reduce program to understand Map Reduce Paradigm.
7. Write a Map Reduce program that mines weather data. Hint: Weather sensors collecting data every hour at many locations across the globe gather

a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented.

8. Implement matrix multiplication with Hadoop Map Reduce
9. Installation of PIG.
10. Write Pig Latin scripts sort, group, join, project, and filter your data.
11. a. Run the Pig Latin Scripts to find Word Count b. Run the Pig Latin Scripts to find a max temp for each and every year
12. Installation of HIVE.
13. Use Hive to create, alter, and drop databases, tables, views, functions, and indexes.

COMPILER DESIGN LAB			
Course Code:	**AI383**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials + Practicals (Hrs/Week):	**00 + 00+ 03**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T+P):	**00 + 00+**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Develop problem solving ability using programming
2. Develop ability to design and analyze a compiler
3. To understand the implementation of lexical analyzer, parse and other compiler design aspects.
4. To write codes for various top down and bottom up parsers and verify them for correctness.
5. To understand Linux Utility Lex and Yacc tools.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand the working of Lex and Yacc compiler for debugging of programs
2. Understand and define the role of lexical analyzer, use of regular expression and transition diagrams.
3. Understand and use Context free grammar, and parse tree construction.
4. Learn & use the new tools and technologies used for designing a compiler
5. Learn how to write programs that execute faster

List of Experiments:

1. Practice of Lex/Yacc of Compiler writing.
2. Write a program to check whether a string belongs to the grammar or not.
3. Write a program to generate a parse tree.
4. Write a program to find leading terminals.
5. Write a program to find trailing terminals.
6. Write a program to compute FIRST of non-terminals.
7. Write a program to compute FOLLOW of non-terminals.
8. Write a program to check whether a grammar is left recursive and remove left recursion.
9. Write a program to remove left factoring.
10. Write a program to check whether a grammar is operator precedent.
11. Construct a recursive descent parser for an expression.
12. Construct a shift reduce parser for a given grammar.
13. Write a program to implement Thomson's construct.
14. Implement intermediate code generation for simple expression

DEEP LEARNING LAB USING PYTHON			
Course Code:	**AI385**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3P**	**Course Semester (U / P):**	**5P**
No. of Lectures + Tutorials (Hrs/Week):	**02 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 To understand the theoretical foundations, algorithms and methodologies of Neural Network.
2 To design and develop an application using specific deep learning models.
3 Training the data sets forms an important part of Deep Learning models.
4 Learn to design Logic Gates using Perceptron.
5. Understand the intuition behind Convolutional Neural Networks and Recurrent Neural Networks.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand the characteristics and types of artificial neural network and Remember working of biological Neuron and Artificial Neural Network.
2 Apply learning algorithms on perceptron and apply back propagation learning on Neural Network.
3 Apply Feedback NN and plot a Boltzmann machine and associative memory on various application.
4 Apply different types of auto encoders with dimensionality reduction and Regularization.
5 Design Convolutional Neural Network and classification using Convolutional Neural Network.

LIST OF EXPERIMENTS:

1. To Write a program to implement Perceptron.
2. To write a program to implement AND OR gates using Perceptron.
3. To implement Crab Classification using pattern net
4. To write a program to implement Wine Classification using Back propagation.
5. To write a MatLab Script containing four functions Addition, Subtraction, Multiply and Divide functions
6. Write a program to implement classification of linearly separable Data with a perceptron
7. To study Long Short Term Memory for Time Series Prediction
8. To study Convolutional Neural Network and Recurrent Neural Network
9. To study ImageNet, GoogleNet, ResNet convolutional Neural Networks
10. To study the use of Long Short-Term Memory / Gated Recurrent Units to predict the stock prices based on historic data.

SEMESTER VI

IoT and its Applications			
Course Code:	**AI302**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Students will be explored to the interconnection and integration of the physical world in IoT.			
2. Learning of networking concepts in IoT environment.			
3. Understanding of various wireless network, topologies, IoT protocols.			
4. Understanding of the importance of security issues in IoT.			
5. Implementation of IoT in real life with learning of tools like MATLAB.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand about all concepts of Internet of Things.			
2. Understand building blocks of Internet of Things and its characteristics.			
3. Learn application protocols for IoT.			
4. Able to understand the application areas of IoT.			
5. Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.			

UNIT I INTRODUCTION TO IOT

Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and OT, IoT Challenges, Drivers Behind New Network Architectures: Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support.

UNIT II IOT NETWORK ARCHITECTURE AND DESIGN

Comparing IoT Architectures: The one M2M IoT Standardized Architecture, The IoT World Forum (IoTWF) Standardized Architecture, Additional IoT Reference Models, A Simplified IoT Architecture, The Core IoT Functional Stack- Layer 1: Things: Sensors and Actuators Layer, Layer 2: Communications Network Layer, Layer 3: Applications and Analytics Layer, IoT Data Management and Compute Stack:Fog Computing , Edge Computing, The Hierarchy of Edge, Fog, and Cloud.

UNIT III NETWORK AND APPLICATION PROTOCOLS FOR IOT

Wireless Communication Technologies: ZigBee, ESP8266, Introduction to sensors and modules - concept, layout, working, applications, Introduction of IoT Development Boards-Node MCU, Arduino, IoT Access Technologies 107IEEE 802.15.4, IEEE 802.15.4g and 802.15.4e, IEEE 1901.2a, IEEE 802.11ah, LoRaWAN, Constrained Devices, Constrained-Node Networks, Optimizing IP for IoT :From 6LoWPAN to 6Lo, Header Compression, Fragmentation, Mesh Addressing, Mesh-Under Versus Mesh-Over Routing, Authentication and Encryption on Constrained Nodes , Application Protocols for IoT: CoAP, Message Queuing Telemetry Transport (MQTT) .

UNIT IV DATA ANALYTICS AND SECURITY OF IOT

An Introduction to Data Analytics for IoT, Structured Versus Unstructured Data, Data in Motion Versus Data at Rest, IoT Data Analytics Overview, IoT Data Analytics Challenges, Machine Learning : Machine Learning Overview Supervised Learning, Unsupervised Learning, Neural Networks, Securing IoT : Common Challenges in IoT Security, Device Insecurity, Network Characteristics Impacting Security, Security Priorities: Integrity, Availability, and Confidentiality, Formal Risk Analysis Structures: IAS OCTAVE, Top Vulnerabilities of Iot.

UNIT VIMPLEMENTING IoT IN REAL LIFE

Interfacing sensors with development boards, communication modules with sensors, communication modules with development boards, MATLAB and Arduino Interfacing, Hands-on in IoT - various real life projects involving different boards, sensors, modules and communication technologies.

Text Books:

1. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things by Rob Barton, Gonzalo Salgueiro, David Hanes
2. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition,
   VPT, 2014.
3. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013

EXPERT SYSTEMS			
Course Code	**AI304**	**Course Credit**	**03**
Course Category	**CC**	**Course(U/P)**	**U**
Course year(U/P)	**3U**	**Course Semester(U/P)**	**6U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**45+00**	**End Term Exam Hours:**	**03**

COURSE OBJECTIVES
1. The objective of this course is to introduce students to the foundation of computability theory.
2. Application of mathematical techniques and logical reasoning to important problem.
3. Develop a strong background in reasoning about finite state automata and formal language.
4. This course is to explore the theoretical foundations of computer science from the perspective of formal language and classify machines by their power to recognize languages.
5. the basic theory of computer science and formal methods of computation like automation theory, formal language, grammars, Turing machine
Course Outcomes
At the end of the course the student should be able to understand the :
1. Under the basic property of regular grammar and design automata
2.Language accepted by an automata i.e. DFA(Deterministic Finite Automata)/NDFA(Non deterministic finite automata).
3. Understand the regular expression(RE) ,Kleen closure ,positive closure, RE to FA and FA to RE
4.Closure property of different language and Decidability /Undesirability property of different languages.
5.Define the various categories of language grammars in the Chomsky hierarchy and variants of Turing machine

UNIT-I

INTRODUCTION: Introduction to AI: Intelligent agents – Perception–Natural language processing – Problem – Solving agents – Searching for solutions: Uniformed search strategies – Informed searchstrategies.

UNIT-2

KNOWLEDGE AND REASONING: Adversarial search – Optimal and imperfect decisions – Alpha, Beta pruning–Logical agents: Propositional logic– First order logic – Syntax and semantics Using first order logic–Inference in first order logic.

UNIT-3

UNCERTAIN KNOWLEDGE AND REASONING Uncertainty – Acting under uncertainty – Basic probability notation – Axioms of probability – Baye's rule – Probabilistic reasoning – Making simple decisions.

UNIT-4

PLANNING AND LEARNING Planning: Planning problem – Partial order planning – Planning and acting in non-deterministic domains – Learning: Learning decision trees – Knowledge in learning – Neural networks – Reinforcement learning – Passive and active.

UNIT-5

EXPERT SYSTEMS: Definition – Features of an expert system – Organization – Characteristics – Prospector – Knowledge Representation in expert systems – Expert system tools–MYCIN–EMYCIN.

TEXT BOOKS:

1. Stuart Russel and Peter Norvig, 'Artificial Intelligence A Modern Approach', Second Edition, Pearson Education, 2003 /PHI.
2. Donald A.Waterman, 'A Guide to Expert Systems', Pearson Education.

REFERENCE BOOKS:

1. George F.Luger, 'Artificial Intelligence – Structures and Strategies for Complex Problem Solving', Fourth Edition, Pearson Education, 2002.

2. Elain Rich and Kevin Knight, 'Artificial Intelligence', Second Edition Tata McGraw Hill, 1995.

CLOUD COMPUTING			
Course Code:	**AI306**	**Course Credits:**	**3**
Course Category:	**E5/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Comprehensive and in-depth knowledge of Cloud Computing concepts,
2. Understand the technologies, architecture and applications
3. Cloud Computing fundamental issues, technologies, applications and implementations.
4. Another objective is to expose the students to frontier areas of Cloud Computing and information systems,
5. while providing sufficient foundations to enable further study and research.

COURSE OUTCOMES
1. At the end of the course the students should be able to:
2. Understand the fundamental principles of distributed computing
3. Understand the importance of virtualization in distributed computing and how this has enabled the development of Cloud Computing
4. Analyze the performance of Cloud Computing.
5. Understand the concept of Cloud Security.
6. Learn the Concept of Cloud Infrastructure Model.

UNIT 1: INTRODUCTION TO CLOUD COMPUTING: Definition, Characteristics, Components, Cloud provider, SAAS, PAAS, IAAS and Others, Organizational scenarios of clouds, Administering & Monitoring cloud services, benefits and limitations, Deploy application over cloud, Comparison among SAAS, PAAS, IAAS Cloud computing platforms: Infrastructure as service: Amazon EC2,Platform as Service: Google App Engine, Microsoft Azure, Utility Computing, Elastic Computing

UNIT 2: INTRODUCTION TO CLOUD TECHNOLOGIES: Study of Hypervisors Compare SOAP and REST Web Services, AJAX and mashups-Web services: SOAP and REST, SOAP versus REST, AJAX: asynchronous 'rich' interfaces, Mashups: user interface services Virtualization Technology: Virtual machine technology, virtualization applications in enterprises, Pitfalls of virtualization Multi Tenant software: Multi-entity support, Multi-schema approach, Multi-tenancy using cloud data stores, Data access control for enterprise applications

UNIT 3: DATA IN THE CLOUD: Relational databases, Cloud file systems: GFS and HDFS, Big Table, HBase and Dynamo. Map-Reduce and extensions: Parallel computing, The map-Reduce model, Parallel efficiency of Map-Reduce, Relational operations using Map-Reduce, Enterprise batch processing using Map-Reduce, Introduction to cloud development,

Example/Application of Mapreduce, Features and comparisons among GFS,HDFS etc, Map-Reduce model Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud Cloud computing security architecture:Architectural Considerations- General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro-architectures; Identity Management and Access control-Identity management, Access control, Autonomic Security.

Cloud computing security challenges: Virtualization security management- virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud

UNIT 4: Issues in cloud computing, Implementing real time application over cloud platform Issues in Intercloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Quality of Service (QoS) monitoring in a Cloud computing environment. Cloud Middleware. Mobile Cloud Computing. Inter Cloud issues. A grid of clouds, Sky computing, load balancing, resource optimization, resource dynamic reconfiguration, Monitoring in Cloud

UNIT 5: Cloud computing platforms, Installing cloud platforms and performance evaluation Features and functions of cloud platforms: Xen Cloud Platform, Eucalyptus, OpenNebula, Nimbus, TPlatform, Apache Virtual Computing Lab (VCL), Enomaly Elastic Computing Platform

Text Books:

1. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, Cloud Computing for Dummies by (Wiley India Edition)
2. Gautam Shroff, Enterprise Cloud Computing by, Cambridge
3. Ronald Krutz and Russell Dean Vines, Cloud Security by, Wiley-India

METAHEURISTICS FOR OPTIMIZATION			
Course Code	**AI 308**	**Course Credits:**	**4**
Course Category	**CC**	**Course(U/P)**	**U**
Course Year(U/P)	**3U**	**Course Semester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/Week)	**03+01**	**Mid Sem.ExamHours:**	**1.5**
TotalNo.of Lectures(L+T)	**45+15**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1. Description of computational intelligence techniques and methods, particularly metaheuristic algorithms for optimization problems.			
2. Knowledge objectives to present basic principles of metaheuristic techniques			
3. Abilitation objectives to identify the techniques appropriate to a given			
4. Attitude objectives to argue the utility of meta heuristic algorithma in solving real-world problems.			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Interpret and explain the concepts of Metaheuristics based optimization and it's application in a diverse range of applications.			
2. Model single solution and population-based Metaheuristic algorithms to solve a given optimization problem.			
3. Model Metaheuristic algorithms to solve Multi-objective optimization problems.			
4. Model hybrid Metaheuristic algorithms to solve a given optimization problem.			
5. Explain algorithms and architectures for parallel implementation of Metaheuristics.			

UNIT I Introduction

Optimization Models, Approximate Algorithms, When to use Metaheuristics?, Methods and Application. Representation, Objective Functions; Constraint Handling; Parameter Tuning; Performance Analysis.

UNIT II Single-Solution Based Metaheuristics/Population-Based Metaheuristics Methiods Single-Solution Based Metaheuristics :Basic Concepts, Fitness Landscape Analysis; Local Search; Tabu Search; Iterated and Guided Local search; Variable Neighborhood Search; Smoothing Methods; Noisy Methods. Population-Based Metaheuristics Methiods : Basic Concepts; Evolutionary Algorithms, Swarm Intelligence, Stochastic diffusion search, Social cognitive optimization

UNIT III Metaheuristics for Multiobjective Optimization/ Fitness Assignment Strategies and Evaluation of Multiobjective Optimization Metaheuristics for MultiobjectiveOptimization : Basic concepts; Multiobjective Continuous and Combinatorial Problems, Multicriteria Decision Making; Design Issues Fitness Assignment Strategies and Evaluation of MultiobjectiveOptimization : Scalar approach, Criterion-Based Methods; Dominance-Based Approaches; Indicator based Approaches; Diversity Preservation; Performance Evaluation

UNIT IV Hybrid Metaheuristics Design and Implementation Issues; Mathematical Programming Approaches; Classical Hybrid Approaches; Hybrid Metaheuristics with Machine Learning and Data Mining; Hybrid Metaheuristics for Multiobjective Optimization

UNIT V Parallel Metaheuristics Parallel Design and Implementation of Metaheuristics; Parallel Metaheuristics for Multiobjective Optimization

TextBooks:

1. Metaheuristics: From Design to Implementation by El-Ghazali Talbi, Wiley, June 2009.
2. Sean Luke, 2013, Essentials of Metaheuristics, Lulu, second edition, available athttp://cs.gmu.edu/□sean/book/metaheuristics.
3. Gandomi, Amir; Yang, Xin-She; Talatahari, Siamak; Alavi, Amir; "Metaheuristic Algorithms in Modeling and Optimization", Metaheuristic Applications in Structures and Infrastructures, Dec 2013.
4. Kalyanmoy Deb; "Multi-Objective Optimization Using Evolutionary Algorithms:An Introduction"; https://www.egr.msu.edu/~kdeb/papers/k2011003.pdf
5. Kalyanmoy Deb; "Single and Multi-Objective Optimization Using Evolutionary Algorithms"; https://www.iitk.ac.in/kangal/papers/2004002.pdf 6. Paulo Corte

ELECTIVE 3

BIOMETRIC SECURITY			
Course Code	**AI310**	**Course Credits**	**3**
Course Category	**CC**	**Course(U/P)**	**U**
Course Year(U/P)	**3U**	**Course Semester(U/P)**	**6U**
No. of Lectures + Tutorials (Hrs/Week)	**03+00**	**Mid Sem. Exam Hours**	**1.5**
Total No. of Lectures(L+ T)	**45+00**	**End Sem. Exam Hours**	**3**
COURSEOBJECTIVES			
1.To provide students with understanding of biometrics.			
2.Make aware of different types of biometrics devices.			
3.Learn different process equipment and their working.			
4.Understanding of Security and standards applied to the security.			
5.To understand attacks in security from malicious attackers.			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1.Demonstrate knowledge of the basic physical and biological science and engineering Principles underlying biometric systems.			
2.Understandandanalyzebiometricsystems at the component level and be able to analyze And design basic biometric system application.			
3.Be able to work effectively in teams and express their work and ideas orally and in writing..			
4.Identifythe sociological and acceptance issues associated with the design and Implementation of biometric systems.			
5.UnderstandvariousBiometricsecurityissues.			

UNIT I INTRODUCTIONTOBIOMETRICS

Introduction- benefits of biometrics over traditional authentication systems -benefits of biometrics identification systems-selecting a biometric for a system –Applications - Key biometric terms and processes-biometric matching methods-Accuracy in biometric systems.

UNIT II PHYSIOLOGICALBIOMETRICTECHNOLOGIES

Physiological Biometric Technologies: Fingerprints - Technical description –characteristics -Competingtechnologies-strengths–weaknesses–deployment-Facialscan-Technicaldescription-characteristics weaknesses-deployment - Iris scan - Technical description – characteristics - strengths – weaknesses –deployment-Retina vascular pattern

UNIT III MEMORY&STORAGEMANAGEMENT

Technical description – characteristics - strengths – weaknesses – deployment - Hand scan – Technical description-characteristics - strengths – weaknesses deployment – DNA biometrics. Behavioral Biometric Technologies: Handprint Biometrics –DNA Biometrics.

UNIT IV SIGNATUREANDHANDWRITINGTECHNOLOGY

Signature and handwriting technology - Technical description – classification – keyboard / key stroke dynamics- Voice – data acquisition - feature extraction - characteristics - strengths – weaknesses - deployment.

UNIT V MULTI BIOMETRICS

Multi biometrics and multi factor biometrics - two-factor authentication with passwords - tickets and tokens–executive decision- implementation plan.

GAMING			
Course Code:	**AI 312**	**Course Credits:**	**3**
Course Category: CC	**CC**	**Course(U/P)**	**U**
Course Year(U/P):U	**3U**	**Course Semester(U/P):**	**6U**
No. of Lectures +Tutorials(Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures(L+T):	**45+00**	**End Sem. Exam Hours:**	**3**

COURSEOBJECTIVES
1. To knowBasic prior knowledge of game development and C is recommended, but not required.
2. To know introduction to computers and programming.
3. To know operators and data types in c++
4. To know different types of programming languages
5. To explore statement and arithmetic operator.

COURSEOUTCOMES
At the end of the course the students should be able to:
1. Explain the difference types of programming languages
2. Basic introduction C++ and visual studio, variables
3. Explain use of C# and C ++
4. Explain the application of object oriented programming.
5. Apply AI techniques for Gaming with colour and sound effects

UNIT I INTRODUCTION TO COMPUTER AND PROGRAMMING

Introduction to computers and programming, Operators and data types in C++, History of Computers, Computer Hardware, Different types of Programming Languages, Introduction to C# programming, Constants and variables, Integers, Floats and Strings, statements, Identifying keywords, Examining arithmetic operators

UNIT II INTRODUCTION TO C++ AND VISUAL STUDIO

Basic introduction C++ & Visual Studio, variables, manipulating variables, adding clouds a tree and buzzing bee, random numbers making decisions with if & else, decision, operators, string, SFML time player input.

UNIT III FUNDAMENTAL OF C++

Loops, array and list parameters function. Returning value from functions switch concepts enumeration and functions implementing game mechanics, detecting collisions, adding all the texts and HUD objects. Sound effect and end condition making the game playing, games stores and manipulate constants.

UNIT IV OBJECT ORIENTED PROGRAMMING

Basics of C++ such as OOP, pointers, introduction to the Standard Template Library, and finally building. OpenGL- powered SFML fully-playable complex games. Basic concepts of oops, collisions, sound effect, references and sprite sheet and vertex array,

UNIT V GAME DESIGN PRINCIPLES & POINTER, TEMPLATE

Game design Principles of game design, Game Design Theory, MDA, 8 type of Fun in Game, Visual style, Gameplay.Introduction of pointers and the standard templates library, texture

holding management, coding the bullets class, making the carrying bullets, detecting collisions pickup and bullets

TextBooks:

1- The Art of Game Design: A Book of Lenses, Second Edition Level Up!
The Guide to Great Video Game Design Theory of Fun for Game Design
2- A Complete Guide to Programming in C++ – 1st Edition(Ulla Kirch-Prinz)
3- Beginning C++ through Game Programming – 3rd Edition
(Michael Dawson)

ONLINE RESOURCES

Learning path: C++ Game Programming

KNOWLEDGE ENGINEERING			
CourseCode:	**AI 314**	**CourseCredits:**	**3**
CourseCategory:CC	**CC**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/ Week):	**03+00**	**MidSem.ExamHours:**	**1.5**
TotalNo.of Lectures(L+T):30	**45+00**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1. To explore the practical application of intelligent technologies into the different domains			
2. To give students insight and experience in key issues of data and knowledge processing			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Understand and describe the concepts central to the creation of knowledge bases and expert systems.			
2. Conduct an in-depth examination of an existing expert system with an emphasis on basic methods of creating a knowledge base.			

UNIT 1 Introduction

Overview of data. Information and knowledge**,** Knowledge engineering and Knowledge management**,** Artificial intelligence use in knowledge Engineering**,** Knowledge based system and its applications

UNIT 2 Knowledge Acquisitions

Information gathering**,** Information retrieval**,** Applications of Natural Language processing**,** Morphology, lexicon, syntax and semantics**,** Parsing, POS tagging, named entity tagging

UNIT 3 Machine Learning

Machine Learning and its applications**,** Supervised and unsupervised learning**,** Classification and clustering**,** Classification algorithms**:** Linear classifiers**,** Nearest neighbor**,** Support Vector Machines**,** Decision tree**,** Random forest**,** Neural networks**,** Case based reasoning

UNIT 4 Knowledge Representations and Reasoning

Proposition logic, predicate logic and reasoning, Knowledge representation languages, Non-monotonic reasoning, Probabilistic reasoning

UNIT 5 Ontology Engineering/Knowledge Sharing

Overview to Ontology, Classifications of ontology, Methodology use in Ontology, Ontology VS Language,

Information Distribution and Integration, Semantic web and its applications, RDF and linked data, Description logic, Web Ontology language, Social web and semantics

TEXT BOOK:

1. Kendal, Simon, Creen, Malcolm, An Introduction to Knowledge engineering, Springer first edition, 2007

2. R.J. Brachman and H.J. Levesque. Knowledge representation and reasoning (Elsevier 2004)

REFERENCES

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A modem approach (Prentice Hall edition , second edition, 2002)
2. P. Jackson, Introduction to expert systems, Addison Wesley, 1999.
3. John Debenham, Knowledge Engineering: Unifying Knowledge Base and Database Design, Springer, 1998.
4. Dan W.Patterson, "Introduction to Artificial Intelligence and Expert Systems", Prentice Hall India Ltd., New Delhi, 2009, ISBN: 81-203-0777-1.
5. Rajendra Akerkar, Introduction to Artificial Intelligence, PHI Learning Pvt. Ltd., 2005, ISBN: 81-203- 2864-7.

PREDICTIVE ANALYSIS			
Course Code:	**AI 316**	**Course Credits:**	**3**
Course Category: CC	**CC**	**Course(U/P)**	**U**
Course Year(U/P):U	**3U**	**Course Semester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem. Exam Hours:**	**1.5**
TotalNo.of Lectures(L+T):30	**45+00**	**EndSem. Exam Hours:**	**3**

COURSEOBJECTIVES
1. To know regression methods
2. To know applications for model building prototyping and full scale the logical implications.
3. To know the implementation of forecasting of inventory models.
4. To know about managing resources, setting ticket prices.
5. To explore managing equipment maintenance, developing credit risk models.
COURSEOUTCOMES
At the end of the course the students should be able to:
1. Financial services to aerospace.
2. Linear regression models and & least squares, multi regression.
3. Explain classification trees and boosting.
4. Explain reproducing kernels. SVM for classification
5. Numerical optimization, boosting methods.

UNIT I LINEAR METHODS OF REGRESSION AND CLASSIFICATION

Overview of supervised learning, Linear regression models and least squares, Multiple regression, Multiple outputs, Subset selection , Ridge regression, Lasso regression, Linear Discriminant Analysis, Logistic regression, Perceptron learning algorithm.

UNI-II MODEL ASSESMENT AND SELECTION

Bias, Variance, and model complexity, Bias-variance tradeoff, Optimism of the training error rate, Estimate of In-sample prediction error, Effective number of parameters, Bayesian approach and BIC, Cross- validation, Boot strap methods, conditional or expected test error

UNIT-III ADDITIVE MODELS, TREES AND BOOSTING

Generalized additive models, Regression and classification trees ,Boosting methods-exponential loss and Ada Boost, Numerical Optimization via gradient boosting ,Examples (Spam data, California housing , New Zealand fish, Demographic data)

UNIT IV NEURAL NETWORK (NN) AND SUPPORT VECTOR

Introduction of Neural network and Support Vector Machines (SVM), and K-nearest Neighbor: Fitting neural networks, Back propagation, Issues in training NN, SVM for classification, Reproducing Kernels, SVM for regression, K-nearest –Neighbor classifiers(Image Scene Classification)

UNIT -V UNSUPERVISED LEARNING AND RANDOM FORESTS

Unsupervised Learning and Random forests: Association rules, Cluster analysis, Principal Components, Random forests and analysis.

TextBooks:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, the Elements of Statistical Learning-DataMining,Inference,andPrediction, SecondEdition, SpringerVerlag, 2009

REFERENCES

1-**Annase Barrie**: Predictive Analytics for Dummies, 2013
2. **Steven Finlay**: Predictive Analytics and Data Mining 2014

Digital Fabrication			
Course Code:	**AI 318**	**Course Credits:**	**3**
Course Category: CC	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03+ 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):30	**45+ 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To know to strategies and techniques for using computers in combination with traditional and analog fabrication processes to shape physical materials and make things.			
2. To know applications for model building prototyping and full scale the logical implications.			
3. To know implications of automation and digital technology for design.			
4. To know about manufacturing, labor, craft, and material culture.			
5. To explore artificial intelligence techniques CAD.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Explain the 3D Computer Aided Design (CAD)These techniques have fabrication relevant for a range of disciplines and			
2. Apply Computer Aided Machining (CAM), and Computer Numeric Controlled (CNC) Machining including techniques for solving uncertainty problems.			
3. Explain use CAD and CNC.			
4. Explain and apply probabilistic models for various use cases.			
5. Apply AI techniques for 3D.			

UNIT I INTRODUCTION ABOUT 2D

Basic introduction to Computer Aided Design Software and Workspace Introduction / 2D CAD – Review syllabus, quick introduction and overview, overview - 2D CAD sketch format / planes, line types, editing, patterns, smart dimensions, sketch relations,Constraints, Introduce design concepts and methodologies

UNI-II DIGITAL TOOLS

Digital tools overview / 2D CAD expanded / 3D basic – Expand on 2D CAD functions, sketch constraints, dimensioning & export for laser cutter, start 3D CAD basic – extrude (and drafted), extrude cut, shell, edit feature (feature tree), additive / subtractive features, approaches / strategies.

UNIT-III INTRODUCTION ABOUT 3D

3D CAD (basic part / mule-body part) - 3D CAD (single part construction & basic drawing) extrude, cut, solid edits (fillets / chamfer), revolve, modify feature (feature tree), basic drawing creation (Legos), basic evaluate (measure), revolve cut. 3D CAD (multi-body part construction) -3D CAD (continued) – combine solids (add / subtract), revolve continued, sweep, dome, move, copy, scale, patterns, mirror, shell, multi-body parts.

UNIT IV ASSEMBELIES AND DRAWING

3D CAD (continued) – revolve review, grip cuts, loft continued, split parts (shelled), assemblies, exploded views, assembly drawing layout, BOM's, part properties, 3D CAD (drawings, analysis, & renderings), basics of rapid prototyping

UNIT -V ADVANCE MODELING

Surface modeling approaches to solids modeling integration techniques, Adobe (Photoshop, Illustrator, In Design. functionality / application for CAD renderings. Fundamentals of graphics, layout for ID presentation - Basic overview / functionality / application – poster, presentation, book.

Text Books:

1. Nick Dunn, "Artificial Intelligence: Digital Fabrication", **Laurence King Publishing (19 September 2012)**

REFERENCES

1-Lisa Iwamoto, Digital Fabrications: Architectural Materials and Techniques (Princeton Architectural Press, 2009)
2-Christopher Boerkrem, Material Strategies in Digital Fabrication(Routledge, 2013)
3-Malcolm McCullough,Abstracting Craft: The Practiced Digital Hand (The MIT Press, 1998)
4-Richard Sennett,The Craftsman (Yale University Press, 2009)
5-Lars Spuybroek,The Sympathy if Things: Ruskin and the Ecology of Design (Bloomsbury, 2012)

ONLINE RESOURCES

Jochen Gros, 50 Digital Wood Joints (http://www.flexiblestream.org/project/50-digital-wood- joints)
David Schultze, Rhino 5 Essential Training (lynda.com) Mode Lab, The Grasshopper Primer,

ELECTIVE 4

AI ENABLED CYBER SECURITY			
Course Code:	**AI320**	**Course Credits:**	**3**
Course Category:	CC	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To prepare students with the technical knowledge and skills needed to protect and defend computer systems and networks.			
2. To develop graduates that can plan, implement, and monitor cyber security mechanisms through AI to ensure the protection of information technology assets.			
3. Analyze and resolve security issues in networks and computer systems using AI and Machine Learning models.			
4. Evaluate and communicate the human role in security systems with an emphasis on ethics, social engineering vulnerabilities and training			
5. Infuse AI capabilities when building smart defensive mechanisms			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Follow a structured model in Security Systems using the concepts in AI and ML			
2. Plan, implement and audit security in a networked, multi-platform and cross platform environment			
3. Protect data and respond to threats through various AI and Machine learning models.			
4. Ability to apply AI and machine learning models in cyber security issues.			
5. Detect attack methodology and combat hackers from intrusion or other suspicious attempts at connection to gain unauthorized access to a computer and its resources			

UNIT-I Introduction to AI for Cyber security:

Applying AI in cyber security, The evolution from expert systems to AI, The different forms of automated learning, characteristics of algorithm training and optimization, AI in Cyber Security and Security Framework: Artificial Intelligence in Cyber Security, Challenges and Promises, Security Threats of Artificial Intelligence

UNIT-II Role of AI in Cyber Security:

Arsenal Classification, Regression, Dimensionality reduction, Clustering, Speech recognition, Video anomaly detection, Natural language processing, Large-scale image processing, Social media analysis

UNIT-III Detecting Cyber security Threatswith AI:

Detect spam with Perceptrons, Image spam detection with support vector machines (SVMs), Phishing detection with logistic regression and decision trees, Spam detection with Naive Bayes, Spam detection adopting NLP.

UNIT-IV Protecting Sensitive Information and Assets:

Authentication abuse prevention, Account reputation scoring, User authentication with keystroke recognition, Biometric authentication with facial recognition.

UNIT-V Fraud Prevention with AI Solutions:

AI and ML algorithms for fraud detection, How bagging and boosting techniques can improve an algorithm's effectiveness, Model Stealing & Watermarking, Network Traffic Analysis, Malware Analysis.

Reference Books:-

1. Leslie F. Sikos, AI in Cybersecurity, Springer Cham,Springer Nature Switzerland AG 2019
2. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
3. Gupta, Brij B., and Quan Z. Sheng, eds. Machine learning for computer and cyber security: principle, algorithms, and practices. CRC Press, 2019.
4. Artificial Intelligence and Data Mining Approaches in Security Frameworks Editor(s): Neeraj Bhargava, Ritu Bhargava, Pramod Singh Rathore, Rashmi Agrawal, 2021.
5. Tsai, Jeffrey JP, and S. Yu Philip, eds. Machine learning in cyber trust: security, privacy, and reliability. Springer Science & Business Media, 2009.
6. Machine Learning: A Probabilistic Perspective, Kevin P Murphy, MIT Press. 6. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer 2006
7. Dr. Nilakshi Jain, Artificial Intelligence, As per AICTE: Making a System Intelligent, Wiley Publication (2019).
8. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies, Packt Publication (2019).

FUZZY LOGIC			
Course Code:	**AI 324**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To develop the fundamental concepts such as fuzzy sets, operations and fuzzy relations			
2. To lean about the fuzzification of scalar variables and the defuzzification of membership functions			
3.To learn three different inference methods to design fuzzy rule based system.			
4. To develop fuzzy decision making by introducing some concepts and also Bayesian decision methods			
5. To learn different fuzzy classification methods.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand the basic ideas of fuzzy sets, operations and properties of fuzzy sets and also about fuzzy relations.			
2. understand the basic features of membership functions, fuzzification process and defuzzification process			
3.design fuzzy rule based system.			
4. know about combining fuzzy set theory with probability to handle random and non-random uncertainty, and the decision making process			
5.Gain the knowledge about fuzzy C-Means clustering			

UNIT-I - Introduction to Fuzzy Sets

Overview of crisp sets; crispness, vagueness, fuzziness and uncertainty; Fuzzy-sets – basic types and basic concepts; α-cuts, strong α-cuts, Representation of fuzzy sets; extension of fuzzy sets.

UNIT-II - Fuzzy Set Operations and Fuzzy Arithmatic

Fuzzy Complement; Fuzzy interaction, t-norms; Fuzzy unions, t-conorms; Combination of operations; Aggregation operation; Fuzzy numbers; Linguistic variables; Arithmatic operations on intervals; Arithmatic Operations on Fuzzy numbers;

UNIT-III - Fuzzy Relations and Fuzzy Logic

Crisp vs Fuzy relations; Projections and Cylindrical extansions; binary fuzzy relations; Binary relations on a single set; Fuzzy equivalence relations; Fuzzy Compatibility Relations; Fuzzy ordering Relations; Fuzzy Morphisms.

Multivalued logics; Fuzzy propositions; Fuzzy quantifiers; Linguistic Hedges.

UNIT-IV - Possibility Theory and Uncertainity-Based Information

Fuzzy measures; Evidence Theory; Possibility Theory; Fuzzy Sets and Possibility Theory; Possibility Theory vs probability Theory.

Information and uncertainty; Nonspecificity of Crisp Sets; Nonspecificity of Fuzzy Sets; Fuzziness of Fuzzy sets.

UNIT-V - Fuzzy Systems and Applications

Membership Functions; Features of the Membership Functions; Fuzzification; Defuzzification to crisp sets; λ-cuts for Fuzzy Relations; Defuzzificationto Scalars;

Fuzzy inference systems; Mamdani's fuzzy models; Sugeno's fuzzy models; Tsukamoto's fuzzy models; other variants; Applications

Reference Books

1. Fuzzy Logic Theory and Applications. Lotfi A Zadeh et al., World Scientific Book, 2018.
2. Introduction To Type–2 Fuzzy Logic Control: Theory and Applications (IEEE Press Series on Computational Intelligence)" by Jerry Mendel and Hani Hagras
3. "Fuzzy Logic and Its Application to Technology and Management" by Dhruba Chakraborty and S. Nanda

DISTRIBUTED OPERATING SYSTEMS			
Course Code	**AI326**	**Course Credits:**	**3**
Course Category	**CC**	**Course(U/P)**	**U**
Course Year(U/P)	**3U**	**Course Semester(U/P):**	**6U**
No.ofLectures+Tutorials (Hrs/Week)	**03+00**	**MidSem. Exam Hours:**	**1.5**
Total No.of Lectures(L+T):30	**45+00**	**End Sem. Exam Hours:**	**3**
COURSEOBJECTIVES			
1.Understand how distributed Operating System is important for Computer System.			
2. Distinguish between centralized systems and distributed systems.			
3.Learn Resource Management in DOS			
4.KnowMemory coherence management			
5.Understandingconsistency model in uniprocessor system and multiprocessing environment			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Gain knowledge of distributed operating system architecture.			
2.Illustrate principles and importance of distributed operating system			
3. Implement distributed client server applications using remote method invocation			
4. Create stateful and state-less applications.			
5. Distinguish between centralized systems and distributed systems.			

UNIT I INTRODUCTION

Introduction of Distributed Operating System (DOS), Functions of DOS, Basic concepts, goals & challenges of distributed systems, architectures of DOS. Revisit the inter process communication

UNIT II COMMUNICATION IN DOS

Communication in DOS : Study of case studies for distributed environment, Issues in communication, message-oriented communication, remote procedure call, remote method invocation, stream-oriented communication, communication between processes, unstructured Vs structured communication, blocking Vs non-blocking communication..

UNIT III SYNCHRONIZATION

Introduction of synchronization, Clocks, events, Time in distributed systems 1. Cristian's algorithm The Berkeley Algorithm, Network Time Protocol (NTP),Logical time and logical clocks,Lamport logical clock,vector clock.

UNIT IV TRANSACTION AND CONCURRENCY CONTROL

Basic concurrency control mechanism in DOS mutual exclusion in distributed environment, Transactions and Concurrency Control in distributed environment, distributed deadlocks in distributed environment.

UNIT V DISTRIBUTED AND SHARED MEMORY MANAGEMENT(DSM):

Basic fundamentals of shared memory in DOS, Architecture and algorithm of distributed shared memory, advantages & challenges of DSM, Memory coherence, consistency model, consistency with uniprocessor system, consistency with multiprocessing environment.

TextBooks:

1. Andrew S. Tanenbaum & Maarten van Steen, Distributed Systems: Principles and Paradigms, Prentice-Hall(2002) ISBN0-13-088893-1
2. D. L. Galli, Distributed Operating Systems, Prentice-Hall(2000) ISBN0-13-079843-6
3. Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez, Prentice Hall International
4. Distributed Operating Systems and Algorithms, Randy Chow, T. Johnson, Addison Wesley
5. Distributed Systems Concepts and Design, G. Coulouris, J. Dollimore, Addison Wesley

ReferenceBooks:

[1]JamesLPeterson,OperatingSystemsConcept,JohnWiley&SonsInc,the6Revedition,2007
[2] Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
[3] StallingsWilliam,OperatingSystems,PHI,NewDelhi,1997.
[4] S. Tanenbaum Modern Operating Systems, Pearson Education, 3rd edition, 2007.
[5] Nutt,OperatingSystem,PearsonEducation,2009.
[6] S.Tanenbaum,DistributedOperatingSystems,PrenticeHall,2nd edition

BUSSINESS ANALYTICS			
Course Code:	**AI328**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P)**	**U**
Course Year(U/P):	**3U**	**Course Semester(U/P):**	**U**
No. of Lectures+Tutorials(Hrs/Week):	**3+00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures(L+T)	**45+00**	**End Sem.Exam Hours:**	**3**
COURSEOBJECTIVES			
1.Ability to apply course concepts to real business problems.			
2. focuses on honing your understanding of key concepts			
3. managerial judgment			
4. To impart knowledge about business analytics, business management, and key analytical skills to make business decisions			
5. Focuses on key business and economical concepts through the study			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Understanding of both national and international business and economics.			
2. Knowledge of overall business analytics through the subjects like organizational behavior			
3. data management, business intelligence, financial management			
4. Focuses include empirical analysis, analytical modeling, and methodology development.			
5. Suggest supply chain management business analytics operations and project management.			

Unit-I Describing and Summarizing Data

Recognize trends in data and detect outliers, Summarize data sets concisely ,Analyze relationships between variables, Create visual representations of data in Excel ,Define and calculate descriptive statistics, Create scatter plots and calculate the correlation coefficient

Unit II Sampling and Estimation

Create representative samples and draw conclusions about the larger population, Craft sound survey questions, Calculate sample statistics and apply the properties of the normal distribution,Calculate confidence intervals to estimate the accuracy of statistics

Unit III: Hypothesis Testing

Quantify the evidence in favor of or against your hypothesis in order to make managerial decisions ,Develop and test hypotheses in Excel to assess the impact of changes on an entire population or estimate differences between populations, Interpret the results of a series of website A/B tests

Unit IV Single Variable Linear Regression

Analyze the relationship between two variables and develop forecasts for values outside the data set ,Identify the best fit line for a data set and interpret its equation through an analysis of housing data, Perform a regression analysis of box office and home video sales using Excel and interpret the output

Unit V Multiple Regression

Identify relationships among three or more variables to improve understanding of data and

provide better forecasts , Estimate the relative predictive power of different combinations of variables by performing and interpreting a multiple variable regression analysis using Excel ,Apply multiple regression analysis to a staffing challenge faced by a hotel , Expand the range of your analysis by using dummy and lagged variables

Texts Books

1. Microsoft Excel Data Analysis and Business Modeling by Wayne L. Winston
2. Naked Statistics: Stripping the Dread from the Data by Charles Wheelan
3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney
4. SQL in 10 Minutes a Day by Ben Forta
5. Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole NussbaumerKnaflic
6. Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau by Ryan Sleeper
7. The Hundred-Page Machine Learning Book by Andriy Burkov
8. The Pyramid Principle: Logic in Writing and Thinking by Barbara Minto
9. Scoring Points: How Tesco Continues to Win Customer Loyalty by Terry H

INTERNET OF THINGS LAB			
Course Code:	**AI382**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Labs (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understanding of all basics of sensor networks in IoT environment.
2. Learning of network simulators.
3. Understanding of creation of various wireless network and topologies.
4. Understanding of the importance of energy usage in IoT applications.
5. Understand the performance sensors in IoT environment.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand about Internet of Things applications.
2. Understand the network technology 6LoWPAN.
3. Learn network simulator COOJA, operating system Contiki, and VMWare.
4. Learn to simulate various networks using different sensors.
5. Learn about role of energy reduction in IoT applications.

List of Experiments

1. Basics of sensor networks, IoT, 6LoWPAN nodes (used in IoT applications), OS Contiki, Network Simulator COOJA.
2. Download and Installation of Contiki (OS for IoT), Creation of Virtual Machine, Download and Installation of VM Player.
3. Initialization of Network Simulator COOJA, understanding of all windows on simulator, study the Mote Configuration, Program the Motes so that all motes display "Hello World" on the output window, Change the values in files to display any desired output by all the motes.
4. Create a network topology having 5 motes of similar configuration. Program them to broadcast the data. Capture the broadcasted packets and analyze the values of various headers like IPv6, using analyzer. Repeat the program by changing the transmission range of all motes and observe the effect.
5. Create a complete wireless sensor network (WSN) topology having 6 motes. Configure 1 mote as Border Router and rest of the 5 motes as sender Motes. Go to the browser and check for the values of routing table of your WSN.
6. Repeat the above program on different topology (some motes should not be in the direct range of border router) and check its effect on routing table of border router.
7. Create a Client-Server network topology having 8 motes. Configure 2 motes as server and 6 motes as client. Capture the packets and generate its pcap files. Analyze the captured packets using packet analyzer tool Wireshark.
8. Create a wireless sensor network (WSN) topology having 20 motes of Z1 type. Configure 3 motes as Sink motes and 17 motes as sender motes. Capture the packets

in using 6LoWPAN analyzer and also in Wireshark. Analyze the various values of captured packets.

9. Create a wireless sensor network (WSN) topology having 15 motes of Z1 type. Configure 2 motes as Sink motes and 13 motes as sender motes. Calculate complete power consumption of all the motes.
10. Repeat above program on any desired topology. Observe the values of power consumption when motes are in Transmit mode, Receive mode, Sleep mode and Processing mode. Also Calculate power consumption of complete network.

EXPERT SYSTEM LAB			
Course Code:	**AI 384**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Labs (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Understand the basics of State space search problems (Hill Climbing, Heuristics search etc.) in computer science.			
2. Understand the various types of Knowledge Representation technique..			
3. Understand the relationship between monotonic & non monotonic reasoning.			
4. Understand advanced learning techniques.			
5. Solve various problems using Expert system			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Identify various networks and learning algorithms in artificial neural network.			
2. Define Fuzzy set, rules and membership function and also defuzzification for a given problem. 3.Identify areas of application for Expert Systems.			
3. Apply the concepts of ANN and Fuzzy Logic in solving engineering problems and implementing controllers.			
4. Discuss various concepts of Genetic Algorithm, Identify various hybrid control strategies.			
5. Identify various networks and learning algorithms in artificial neural network.			

1. Design and Development of an expert system which incorporate following programs-
   I. Feasibility study ; to be developed expert system.
   II. Write the If-Then rules for the development of expert system.
   III. Development of database and relations with proper data types.
   IV. Write a program for forward chaining mechanism.
   V. Write a program for backward chaining mechanism.
   VI. Write a program for Hybrid chaining mechanism.
   VII. Write a program for the connection establishment between front-end and back-end.
2. Study of different commercial expert system shells.
3. To Study JESS expert system
4. To Study RVD expert system

CLOUD COMPUTING LAB			
Course Code:	**AI 386**	**Course Credits:**	**2**
Course Category:	**CC**	**Course(U/P)**	**U**
Course Year(U/P):	**3U**	**Course Semester(U/P):**	**6U**
No. of Lectures +Tutorials(Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures(L+T):	**10+00**	**End Sem. Exam Hours:**	**3**
COURSEOBJECTIVES			
1.To develop web applications in cloud			
2.To learn the design and development process involved in creating a cloud-based application			
3.To learn to implement and use parallel programming using Hadoop			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1. Configure various virtualization tools such as Virtual Box, VMware workstation.			
2. Design and deploy a web application in a PaaS environment.			
3. Learn how to simulate a cloud environment to implement new schedulers.			
4. Install and use a generic cloud environment that can be used as a private cloud.			
5. Manipulate large data sets in a parallel environment.			

LIST OF EXPERIMENTS

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

SEMESTER VII

Parallel Processing and CUDA Programming			
CourseCode:	**MA 402**	**CourseCredits:**	**4**
CourseCategory:CC	**CC**	**Course(U/P)**	
CourseYear(U/P):U	**4U**	**CourseSemester(U/P):**	**U**
No.ofLectures+Tutorials Hrs/Week):	**3+01**	**MidSem.ExamHours:**	**1.5**
TotalNo.of Lectures(L+T):30	**45+15**	**EndSem.ExamHours:**	**3**
COURSEOBJECTIVES			
1. Define terminology commonly used in parallel computing, such as efficiency and speedup.			
2. Describe common GPU architectures and programming models			
3. Implement efficient algorithms for common application kernels, such as matrix multiplicatio			
4. Given a problem, develop an efficient parallel algorithm to solve it.			
5. Given a problem, implement an efficient and correct code to solve it, analyze its performanc nd give convincing written and oral presentations explaining the achievements			
COURSEOUTCOMES			
At the end of the course the students should be able to:			
1.Understand the distributed and parallel computing systems			
2. Familiar with parallel and distributed languages MPI, Pthread, OpenMP, and CUDA			
3. Design parallel and distributed algorithms using these parallel languages			
4. Writing Parallel Programs.			
5. Able to meaure performance metrics			

UNIT-1 INTRODUCTION

Introduction to Parallel Computers ,Message-Passing Computing and Programming, Multithread Programming , Parallel Programming Paradigms, Parallel Architecture, Parallel Architecture (case studies)

UNIT-2

CUDA Programming ,Open MP Programming , Embarrassingly Parallel Computation, GPU-Compute Architecture, CUDA, Memory organization in CUDA Multi-Core CPU programming, MPI, PVM, Performance evaluation and scalability

UNIT-3

Partitioning and Divide-and-Conquer Strategies, Pipelined Computation ,Synchronous Computations, Pipelining and Throughput Latency and Latency hiding

UNIT-4

Synchronous Computations, Load Balancing and Termination Detection,Distributed Shard Memory, Synchronization: Memory Consistency, Barriers (local versus global), Atomics, Memory fence. Prefix sum, Reduction. Programs for concurrent Data Structures such as Worklists, Linked-lists. Synchronization across CPU and GPU

UNIT-5

Sorting Algorithms, Min/Max, Sum Searching, Merging, Sorting, Prefix operations N-body problems, Matrix operations

REFERENCE BOOKS:

1. David Kirk and Wen-meiHwu, Programming Massively Parallel Processors: A Hands-On Approach, 2nd Edition, Publisher: Morgan Kaufman, 2012, ISBN: 9780124159921.
2. Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufman; 2012 (ISBN: 978-0124159334
3. An Introduction to Parallel Algorithms by Joseph Jaja (Addison-Wesley Professional)
4. Introduction to Parallel Computing by Ananth Grama, George Karypis, Vipin Kumar and Anshul Gupta (Pearson)
5. Parallel Programaming in C with MPI and openMP by Michael J Quinn (McGraw H

COMPUTATIONAL INTELLIGENCE			
Course Code:	**AI401**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To understand basic techniques of computational models			
2 Understand theoretical model of fuzzy principles.			
3 To focus on how to apply neural network algorithms			
4 To study Soft computing techniques			
5 To study over real-time problems to get optimized outcome.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Recognize and depict soft computing methods and their roles to build intelligent Systems.			
2 Apply fuzzy principles and thinking to deal with vulnerability and tackle real-time Issues.			
3 Apply genetic algorithms to generate optimized results for a particular problem.			
4 Apply neural networks to design classification problems.			
5 Evaluate and compare solutions by various soft computing approaches for a given Problem.			

UNIT I INTRODUCTION
Introduction to CI, History of CI, Basic techniques and applications of CI, Introduction to Neural Network, Fuzzy Logic, Genetic Algorithm, Hybrid System.

UNIT II ELEMENTARY AND ADVANCE SEARCH TECHNIQUES:
State Space Search, Blind Search, Heuristic Search (Hill Climbing, A/A* Algorithm, Min-Max Search, Constraint Satisfaction), Multi-Objective Genetic Algorithm.

UNIT III FUZZY SET THEORY
Fuzzy Sets, Basic Definition and Terminology, Set-theoretic Operations, Member Function Formulation and Parameterization, Fuzzy Rules and Fuzzy Reasoning using IF-THEN rules, Extension Principle and Fuzzy Relations, Fuzzy Inference Systems, Different Fuzzy Models: Mamdani Fuzzy Models, Sugeno Fuzzy Models, Tsukamoto Fuzzy Models, Input Space Partitioning and Fuzzy Modeling.

UNIT IV OPTIMIZATION:
Derivative-based Optimization, Descent Methods, The Method of Steepest Descent, Classical Newton's Method, Step Size Determination, Derivative-free Optimization, Concepts of Genetic Algorithms, GA techniques, Simulated Annealing, Random Search, Downhill Simplex Search, Evolutionary Computing, Swarm optimization, Green Computing, Big data mining

UNIT V NEURAL NETWORKS & DEEP LEARNING:

Artificial Neural Network, Supervised Learning Neural Networks, Perceptrons and its limitations, Adaline, Back propagation learning algorithm MutilayerPerceptrons, Radial Basis Function Networks, Unsupervised Learning Neural Networks, Deep Neural Network, Convolutional Neural Network, Competitive Learning Networks, Kohonen Self- Organizing Networks, Learning Vector Quantization, Hebbian Learning, Hop-field networks.

REFERENCE BOOKS:

1. Fuzzy Logic with Engineering Applications, Timothy J. Ross, McGraw-Hill, 1997.
2. Genetic Algorithms: Search, Optimization and Machine Learning, Davis E. Goldberg, Addison Wesley, N.Y.,1989.
3. Neural Networks: A Comprehensive Foundation, Simon Haykin. Prentice Hall
4. Neural Network Design, M. T. Hagan, H. B. Demuth, Mark Beale, Thomson Learning, Vikash Publishing House.
5. Neural Networks, Fuzzy Logic and Genetic Algorithms, S. Rajasekaran and G.A.V.Pai, PHI, 2003.

Robotics and Drone			
Course Code	**AI403**	**Course Credit**	**02**
Course Category	**CC**	**Course(U/P)**	**U**
Course year	**4U**	**Course Semester(U/P)**	**7U**
No of Lectures + Tutorials(Hrs./Week)	**02+00**	**Mid Semester Exam Hours:**	**1.5**
Total no of Lectures(L+T)	**30+00**	**End Term Exam Hours:**	**03**
COURSE OBJECTIVES			
1. Explain the space of design for robotic communication, safety ,state estimation and control.			
2.Apply the knowledge to construct programs for communication, safety ,state estimation and control.			
3. Apply the knowledge for Robotics			
3. Build, program, and operate an autonomous robot drone.			
4.Hand on experience on design, fabrication and flying of UAV.			
Course Outcomes			
1.students understand about Robotics			
2. The concept of robots ,robot Anatomy, Robot behavior.			
3. Understand the applications of robots in real life applications.			
4. Explain the concepts of Drone and Applications of drones, Propulsion			
5. Describe the parts and functions of UAV & Indian Aviation regulations of UAV			

UNIT I:

History of robots, Classification of robots, Present status and future trends. Basic components of robotic system. Basic terminology- Accuracy, Repeatability, Resolution, Degree of freedom. Mechanisms and transmission ,Design and control issues, Manipulation and control, Sensors and vision, Programming robot, Future aspect.

UNIT II: Drive systems and Sensors

Drive system- hydraulic, pneumatic and electric systems Sensors in robot – Touch sensors, Tactile sensor, Proximity and range sensors, Robotic vision sensor, Force sensor, Light sensors, Pressure sensors.

UNIT III- Kinematics and Dynamics of Robots

2D, 3D Transformation, Scaling, Rotation, Translation, Homogeneous coordinates, multiple transformation, Simple problems. Matrix representation, Forward and Reverse Kinematics Of Three Degree of Freedom, Homogeneous Transformations, Inverse kinematics of Robot, Robot Arm dynamics, D-H representation of robots, Basics of Trajectory Planning.

UNIT IV BASICS OF Drone: Different types of flight vehicles - Components and functions of an airplane - Forces acting on Airplane - Physical properties and structure of the atmosphere , Difference between aircraft and UAV, Parts and functions of Fixed, Rotorcraft and flapping wing UAV – various History of UAV's.

UNIT V-Types and Applications of Drones: Types of Drones, Applications and Uses. Characteristics of Multi rotor vehicle, Fixed Wing vehicle, Flapping wing Vehicles and their applications – Defense, Civil, Environmental monitoring (physical, chemical and biological

Text Books:

1. Mikell P Groover, Nicholas G .Odrey , Mitchel Weiss, Roger N Nagel, Ashish Dutta, "Industrial Robotics, Technology programming and Applications", McGraw Hill, 2012.
2. Craig. J. J. "Introduction to Robotics- mechanics and control", Addison- Wesley, 1999.
3. Andey Lennon " Basics of R/C model Aircraft design" Model airplane news publication

Reference Books:

1. S.R. Deb, "Robotics Technology and flexible automation", Tata McGraw-Hill Education., 2009.
2. Richard D. Klafter, Thomas .A, ChriElewski, Michael Negin, "Robotics Engineering an Integrated Approach", PHI Learning., 2009.
3. Francis N. Nagy, Andres Siegler, "Engineering foundation of Robotics", Prentice Hall Inc., 1987.
4. P.A. Janaki Raman, "Robotics and Image Processing an Introduction", Tata McGraw Hill Publishing company Ltd., 1995.
5. Carl D. Crane and Joseph Duffy, "Kinematic Analysis of Robot manipulators", Cambridge University press, 2008.
6. Fu. K. S., Gonzalez. R. C. & Lee C.S.G., "Robotics control, sensing, vision and intelligence", McGraw Hill Book co, 1987
7. Ray Asfahl. C., "Robots and Manufacturing Automation", John Wiley & Sons Inc.,1985

NATURAL LANGUAGE PROCESSING			
Course Code:	**AI405**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 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. Learns 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.

REFERENCE BOOKS:

1. Daniel Jurafsky , James H. Martin , "Speech & language processing", Pearson publications.
2. Allen, James. Natural language understanding. Pearson, 1995.

ELECTIVE 5

Automation and Robotics			
Course Code:	**AI 407**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P):**	**U**
Course Year(U/P):	**4U**	**Course Semester(U/P):**	**7U**
No. of Lectures +Tutorials (Hrs/Week):	**03+00**	**MidSem. Exam Hours:**	**1.5**
Total No .of Lectures(L+T):	**45+00**	**EndSem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. The subject should enable the students to understand the principles of automation
2. importance of automated flow lines and its types.
3. To learn the concepts of Robotics
4. kineiratics of robot, principles of robot drives and controls
5. sensors used in robots and programming methods.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. what is automation, types of automation, components of automation, strategies rind levels of automation.
2. how the assembly is carried out on automated flow line without interruption and how to balance the line and flexible assembly lines.
3. know the various components in the anatomy of robot. By knowing this the student may apply in the design of new robotic structure.
4. should also learn about the homogeneous transformations and its applications in the analysis of a robotic structure
5. able to understand robot programming languages which may adopt in different applications of robot.

Unit-I

INTRODUCTION TO AUTOMATION: Need, Types, Basic elements of an automated system, Manufacturing Industries, Types of production, Functions in manufacturing, Organization and information processing in manufacturing, Automation strategies and levels of automation. Hardware components for automation and process control, mechanical feeders, hoppers, orienters, high speed automatic insertion devices.

Unit II

AUTOMATED FLOW LINES: Part transfer methods and mechanisms, types of Flow lines, flow line with/without buffer storage, Quantitative analysis of fiow lines.

ASSEMBLY LINE BALANCING: Assembly process and systems assembly line, line balancing methods, ways of improving line balance, flexible assembly lines.

Unit III:

INTRODUCTION TO INDUSTRIAL ROBOTICS: Classification of Robot Configurations, functional line diagram, degrees of freedom. Components common types of arms, joints grippers, factors to be considered in the design of grippers.

ROBOT ACTUATORS AND FEEDBACK COMPONENTS: Actuators, Pneumatic, Hydraulic actuators, Electric &Stepper motors, comparison. Position sensors - potentiometers, resolvers, encoders - velocity sensors, Tactile sensors, Proximity sensors.

Unit IV

MANIPULATOR K4NEMATICS: Homogenous transformations as applicable to rotation and transition - D-H notation, Forward inverse kinematics.

MANIPULATOR DYNAMICS: Differential transformations, Jacobians, Lagrange - Euler and Newton - Euler formations. Trajectory Planning: Trajectory Planning and avoidance of obstacles path planning, skew motion, joint integrated motion - straight line motion.

Unit V

ROBOT PROGRAMMING: Methods of programming - requirements and features of programming languages, software packages. Problems with programming languages. ROBOT APPLICATION IN MANUFACTURING: Material Transfer - Material handling, loading and unloading - Process - spot and continuous arc welding & spray painting - Assembly and Inspection.

TEXT BOOKS:

1. Automation , Production systems and CIM, M.P. Groover /Pearson Edu.
2. Industrial Robotics — Mikell P. Groover and Mitchell Weiss, Roger N. Nagel, Nicholas, G.Odrey — McGraw Hill, 1986..

REFERENCE BOOKS:

1. Robotics and control - R K Mittal and I I nagrath,TataMcGraw Hill 2004.
2. An Introduction to Robot Technology, P. Coiffet and M. Chaironze, Kogam Page Ltd. 1983 London.
3. Robotic Engineering - integrated approach by Richard d Klafter-London: Prentice-Hall- 1989.
4. Robotics, Fundamental Concepts and analysis —AshitaveGhosal, Oxford Press
5. Introduction to Robotics - John J. Craig, PearsonEdu.

Blockchain Technology using SALONA			
Course Code	**AI409**	**Course Credits:**	**3**
Course Category	**CC**	**Course(U/P)**	**U**
Course Year(U/P)	**4U**	**CourseSemester(U/P):**	**7U**
No.ofLectures +Tutorials(Hrs/Week)	**03+00**	**MidSem.Exam Hours:**	**1.5**
Total No. of Lectures(L+ T)	**45+00**	**EndSem.Exam Hours:**	**3**

COURSE OBJECTIVES

1. Blockchain is an emerging technology platform for developing decentralized applications and data storage, over and beyond its role as the technology underlying the cryptocurrencies.
2. The basic tenet of this platform is that it allows to create a distributed and replicated ledger of events, transactions, and data generated through various IT processes with strong cryptographic guarantees of tamper resistance, immutability, and verifiability.
3. The objective of this course is to provide conceptual understanding of how blockchain technology can be used to innovate and improve business processes.
4. The course covers the technological underpinning of blockchain operations in both theoretical and practical implementation of solutions using blockchain technology.

COURSE OUTCOMES:

By the end of this course, the learners will be able to:

1. Explain the fundamental characteristics of blockchain using bitcoin.
2. Demonstrate the application of hashing and public key cryptography in protecting the blockchain
3. Explain the elements of trust in a Blockchain: validation, verification, and consensus.
4. Perform a transaction in bitcoin testnets.
5. Develop smart contracts in Ethereum framework

UNIT 1

Background Theories: Cryptographic Hash Functions (SHA), Cryptographically Secured, Digital Signature, Public Key Cryptography (RSA), Chain of Blocks, Merkle Trees, Smart Contract, Centralized Vs. Distributed network, Actors and components in Blockchain solution, History of Blockchain, the problems that our current system faces, why blockchain, cryptocurrency, real-time blockchain use cases, blockchain overflow, Merkle tree, Genesis Block.

UNIT 2

Introduction to Blockchain: A typical block structure, chain of block, distributed ledger, Permissioned and Permission-less Model, Constructing a chain, Orphan block, Block propagation,understand private blockchain, the difference between private and public blockchain, docker, Hyperledger, Chaincode, CBDC, Corda.

UNIT 3

Understanding of bitcoin and its history, Altcoins, trading, cryptocurrency use cases and the government regulations, Bitcoin: Basics, Creation/Projections of Coins, Double-spending attack, Bitcoin Anonymity, Basics of Bitcoin Script (FORTH), Bitcoin transactions through script.

UNIT 4 Salona Core Concepts Introduction to the smart contracts, increasing in popularity

for a variety of applications - Contract Dimensions - Legal Considerations - Security Challenges & Measures - Smart contract platforms - Smart Contract Implementation Ricardian Contracts & Smart Contracts Practice: Deploy a Smart Contract

UNIT 5 ETHEREUM

An overview of other public networks like Solana, Binance and many more, Deeper knowledge about attacks: Sybil & Phishing, blockchain 2.0 Ethereum, Starting with a brief understanding of Token to Stablecoins, Vesting period, IDO, IEO & DAO, NFT introduction, Solana overview SPL token creation.

Reference Book:

1. Advanced Applications of Blockchain Technology, Shiho Kim, Ganesh Chandra Deka.
2. Blockchain Technology Concepts And Applications, Kumar Saurabh and Ashutosh Saxena.
3. Blockchain Basics: A Non-Technical Introduction in 25 Steps (Daniel Drescher, Apress, 2017)
4. Blockchain Technology and Applications: A systematic and Practical approach by Amit Dua.

3 D Printing			
Course Code:	**AI 411**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P):**	**U**
Course Year(U/P):	**2U**	**Course Semester(U/P):**	**3U**
No. of Lectures +Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures(L+T):	**45+00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To impart students to the fundamentals of various 3D Printing Techniques for application to various industrial needs.			
2.Student will be able to convert part file into STL format			
3.Understand the method of manufacturing of liquid based, powder based and solid based techniques.			
4.To Be able to open, view, manipulate and edit three dimensional object files			
5.Successfully fabricate the file design through a 3D printing service provider using appropriate material and method selections.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1.Use software tools for 3D printing			
2.Prepare 3D printed modules			
3.Construct products using LOM and FDM technologies			
4.Be able to think critically about public reports of 3D printing in the media			
5.Be able to communicate clearly what makes 3D printing unique from other traditional methods of manufacturing and what its current benefits and limitations are.			

Unit I: Introduction

Introduction to Design, Prototyping fundamentals. Introduction to 3D printing, its historical development, advantages. Commonly used terms, process chain, 3D modelling, Data Conversion, and transmission, Checking and preparing, Building, Post processing, RP data formats, Classification of 3D printing process, Applications to various fields.

Unit II: Liquid Based 3D Printing Stereo lithography apparatus (SLA):

Models and specifications, process, working principle, photopolymers, photo polymerization, layering technology, laser and laser scanning, applications, advantages and disadvantages, case studies. Solid ground curing (SGC): Models and specifications, process, working ,principle, applications, advantages and disadvantages, case studies

Unit III: Solid Based 3D Printing Laminated object manufacturing(LOM):

Models and specifications, Process, Working principle, Applications, Advantages and disadvantages, Case studies. Fused Deposition Modeling (FDM): Models and specifications,

Process, Working principle, Applications, Advantages and disadvantages, Case studies, practical demonstration

Unit IV: Principle of FDM/FFF printing

Basic steps to perform FDM printing, Significant process parameters of FDM printing, layer height, raster angle, raster width, build temperature, Nozzle temperature, orientation, printing speed etc

Unit V: Applications of FDM printer in AM

Applications of AM: Aerospace, Biomedical, Automotive, Bio-printing, Tissue & Organ Engineering, Architectural Engineering, Surgical simulation, Art, Health

Text & References

1. Chua C.K., Leong K.F. and LIM C.S Rapid prototyping: Principles an Applications, World Scientific publications, 3rdEd., 2010
2. D.T. Pham and S.S. Dimov, "Rapid Manufacturing", Springer, 2001
3. Terry Wohlers, " Wholers Report 2000", Wohlers Associates, 2000
4. Paul F. Jacobs, " Rapid Prototyping and Manufacturing"–, ASME Press, 1996
5. Ian Gibson, Davin Rosen, Brent Stucker "Additive Manufacturing Technologies, Springer, 2nd Ed, 2014.

PARALLEL DISTRIBUTED SYSTEMS			
Course Code:	**AI413**	**Course Credits:**	**3**
Course Category:	**CC**	**Course(U/P)**	**U**
CourseYear(U/P):	**4U**	**Course Semester(U/P):**	**7U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem. ExamHours:**	**1.5**
TotalNo.of Lectures(L+T):	**45+00**	**EndSem. ExamHours:**	**3**

COURSE OBJECTIVES
1. Understand different parallel and distributed programming paradigms and algorithms, and gain practice in implementing and testing solutions using these.
2. Analyze and critically discuss research papers both in writing and in class.
3. Orally present a clear and accessible summary of a research work
4. Formulate and evaluate a hypothesis by proposing, implementing and testing a project Relate one's project to prior research via a review of related literature
5. Understand the fundamental questions in parallel and distributed computing and analyze different solutions to these questions
COURSE OUTCOMES
At the end of the course the students should be able to:
1. To develop and apply knowledge of parallel and distributed computing techniques and methodologies.
2. To gain experience in the design, development, and performance analysis of parallel and distributed applications.
3. To gain experience in the application of fundamental Computer Science methods and algorithms in the development of parallel applications.
4. To gain experience in the design, testing, and performance analysis of a software system, and to be able to communicate that design to others.
5. Students will examine how existing systems have applied the concepts of distributed systems in designing large systems, and will additionally apply these concepts to develop sample systems

UNIT: -I INTRODUCTION TO PARALLEL DISTRIBUTED SYSTEM

Basic Concepts: Introduction to parallel processing, parallel processing terminology, decomposition, complexity, throughout, speedup, measures, data dependence, resource dependence, Bernstein's conditions levels of parallelism in programs. Program flow-control flow, data flow, Distributed systems – Introduction, advantages, and tightly-coupled loosely-coupled systems. Hardware and software requirements, design issues.

UNIT:-II PARALLEL PROCESSING

Parallel Processing – Structure & Organization: Taxonomy of parallel processes: granularity, basic architectures, multiprocessors, vector processors, pipeline:-both linear as well as non liner pipeline ,optimal design, Arithmetic pipeline, Instruction pipeline, Pipeline hazards and their solution ,reservation table, scheduling.

UNIT: - III INTRODUCTION TO DISTRIBUTED COMPUTING

Distributed Computing-introduction, definition , its history; Distributed Computing system

definition and its evolution, reasons for its popularity, Strength and weaknesses of distributed computing, Different forms of Computing: Minicomputer model, workstation model, workstation server model, Processor pool Model; Cluster:- definitions, reasons for its popularity cluster computer system architecture, Windows cluster, Solaris cluster, Linux cluster; Using cluster, distributed Computing System models: Distributed operating system, Introduction to DCE, architecture of Distributed Applications

UNIT: - IV ANALYTICAL MODELING OF PARALLEL PROGRAMS

Analytical Modeling of Parallel Programs: Sources of Overhead in Parallel Programs, Performance Metrics for Parallel Systems, The Effect of Granularity on Performance, Scalability of Parallel Systems, Minimum Execution Time and Minimum Cost-Optimal Execution Time, Asymptotic Analysis of Parallel Programs, Other Scalability Metrics.

UNIT:-V PARALLEL & DISTRIBUTED PROGRAMMING

Parallel & Distributed Programming: Parallel Programming environments, models, synchronous asynchronous programming, modulla-2, occamm, FORTRAN, DAP FORTRAN, C-linda, Actus, data flow programming, VAL etc,. MPI, Open MP

Text Books:

1. Michael J. Quinn, "Parallel Computing – Theory and Practice, 2nd Edition, McGraw Hill
2. Kai Hwang, "Advanced Computer Architecture – Parallelism, Scalability, Programmability", McGraw Hill Inc, 1993.
3. Wilkinson, "Parallel Programming using networked computer" , Pearson Education India, 20006
4. S. G. Akl, "The Design and Analysis of parallel algorithms",
5. Introduction to parallel computing by Ananth Grama, Anshul Gupta, Gorge Karypis, Vipin Kumar, Pearson.
6. Pradeep K. Sinha," Distributed Systems"

TIME SERIES ANALYSIS AND APPLICATIONS			
Course Code:	**AI415**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course(U/P)**	**U**
Course Year(U/P):U	**4U**	**Course Semester(U/P):**	**7U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem. Exam Hours:**	**1.5**
TotalNo.of Lectures(L+T):	**45+00**	**EndSem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Understand and apply to time series analysis and application.			
2. Able to fit various growth curves, trend and to measure seasonal indices.			
3. Understand forecasting by different methods and to calculate variance of a random component.			
4. Present time series in an informative way, both graphically and with summary statistics,			
5. Model time series to analyses the underlying structure(s) in both the time and frequency domains.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. To develop and apply knowledge to real time series data and interpret outcomes of analyses.			
2. Forecast the trend pattern exhibited by the given data by using various methods			
3. Run and interpret time series models and regression models for time series.			
4. Use the Box-Jenkins approach to model and forecast time series data empirically.			
5. Analyze and estimate the cyclic components using special processes.			

UNIT: -I INTRODUCTION TO TREND

Introduction to times series data, application of time series from various fields, Components of a time series, Decomposition of time series. Trend: Estimation of trend by free hand curve method, method of semi averages, fitting a various mathematical curve, and growth curves.

UNIT: - II TREND AND SEASONAL COMPONENT

Method of moving averages. De-trending. Effect of elimination of trend on other components of the time series. Seasonal Component: Estimation of seasonal component by Method of simple averages, Ratio to Trend, Ratio to moving average and Link relatives.

UNIT:-III DATA ANALYSIS

Data Analysis: Regression modeling, multivariate analysis, Bayesian modeling, inference and Bayesian networks, support vector and kernel methods, analysis of time series: linear systems analysis & nonlinear dynamics, rule induction, neural networks: learning and generalization, competitive learning, principal component analysis and neural networks, fuzzy logic: extracting fuzzy models from data, fuzzy decision trees, stochastic search methods.

UNIT: - IV FORECASTING

Variate component method: Stationary Time series: Weak stationary, auto correlation function and correlogram of moving average .Forecasting: Exponential smoothing methods, Short term forecasting methods: Brown's discounted regression, Box-Jenkins Method.

UNIT:-V CYCLIC COMPONENT

Cyclic Component: Harmonic Analysis. Some Special Processes: Moving-average (MA) process and Auto regressive(AR) process of orders one and two, Estimation of the parameters of AR (1) and AR (2) – Yule-Walker equations.

Text Books:

1. Mukhopadhyay P. (2011): Applied Statistics, 2nd ed. Revised reprint, Books and Allied.
2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer
3. Kendall M.G. (1976): Time Series, Charles Griffin.
4. Frank J Ohlhorst, "Big Data Analytics: Turning Big Data into Big Money", Wiley and SAS Business Series