UNIVERSITY SCHOOL

OF

INFORMATION AND COMMUNICATION TECHNOLOGY

Department Of Computer Science & Engineering

COURSE STRUCTURE

B.TECH
COMPUTER SCIENCE AND ENGINEERING
Specialization: Machine Learning

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	CS101	Fundamentals of Computer Programming	3	1	0	4	CC1/ FC
2	CM101	Fundamentals of Machine Learning	2	0	0	2	CC2/ FC
3	MA101	Engineering Mathematics-I	3	1	0	4	GE1
4	PH102	Engineering Physics	3	1	0	4	GE2
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	CS181	Computer Programming Lab	0	0	2	1	CC-L1/SEC
9	PH104	Engineering Physics Lab	0	0	2	1	GE-L2
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	CM102	Introduction to Python	2	0	0	2	CC3/ FC
2	CM104	Computer Organization & Architecture	3	0	0	3	CC4/ SEC
3	MA102	Engineering Mathematics -II	3	1	0	4	GE4
4	EE102	Basic Electrical Engineering	3	1	0	4	GE4
5	ME101	Engineering Mechanics	3	1	0	4	GE6
6	ES101	Environmental Studies	3	1	0	4	OE2/AECC
7	CM182	Python Programming Lab	0	0	2	1	CC-L2/SEC
9	EE104	Basic Electrical Engineering Lab	0	0	2	1	GE-L4
10	ME102	Workshop Practice	1	0	2	2	GE-L5
11	GP	General Proficiency	Non Credit				

SEMESTER III

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CM201	Internet Technology	3	0	0	3	CC3/ FC
2	CM203	Operating Systems	3	0	0	3	CC4/ SEC
3	CM205	Data Structure & Algorithms	3	0	0	3	GE4
4	CM207	Computer Vision	3	0	0	3	GE4
5	CM209	Introduction to R Programming	3	0	0	3	GE6
6	MA201	Engineering Mathematics-III	3	1	0	4	OE2/AECC
7	CM281	R Programming Lab	0	0	3	2	CC-L2/SEC
8	CM283	Data Structure & Algorithms	0	0	3	2	CC-L3/SEC
9	CM285	Internet Technology Lab	0	0	3	2	GE-L4
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	CM202	Software Engineering	3	0	0	3	CC3/ FC
2	CM204	Database Management System	3	0	0	3	CC4/ SEC
3	CM206	Java Programming	3	0	0	3	GE4
4	CM208	Artificial Intelligence	3	0	0	3	GE4
5	CM210	Theory of Automata	3	0	0	3	GE6
6	CM212	Introduction to MATLAB	3	1	0	4	OE2/AECC
7	CM282	Database Management System Lab	0	0	3	2	CC-L2/SEC
8	CM284	Java Programming Lab	0	0	3	2	CC-L3/SEC
9	CM286	MATLAB	0	0	3	2	GE-L4
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	CM301	Compiler Design	3	0	0	3	CC3/ FC
2	CM303	Soft Computing Techniques	3	0	0	3	CC4/ SEC
3	CM305	Analysis & Design of Algorithms	3	0	0	3	GE4
4	CM307	Big Data Analytics	3	0	0	3	GE4
5	CM309	Machine Learning	3	1	0	4	GE6
6		Elective I	3	0	0	3	OE2/AECC
7	CM381	Analysis & Design of Algorithms Lab	0	0	3	2	CC-L2/SEC
8	CM383	Big Data Analytic sLab	0	0	3	2	CC-L3/SEC
9	CM385	Machine Learning Lab with Python	0	0	3	2	GE-L4
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	CM302	Web Development using PHP	3	0	0	3	CC3/ FC
2	CM304	Deep Learning	3	0	0	3	CC4/ SEC
3	CM306	Reinforcement Learning	3	1	0	4	GE4
4	CM308	Human Machine Interaction	3	0	0	3	GE4
5	CM310	Knowledge Representation	3	0	0	3	GE6
6		Elective2	3	0	0	3	OE2/AECC
7	CM382	Web Development using PHP Lab	0	0	3	2	CC-L2/SEC
8	CM384	Deep Learning Lab using Python	0	0	3	2	CC-L3/SEC
9	CM386	Reinforcement Learning lab using Python	0	0	3	2	GE-L4
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

- Industrial training will be done after third year during the summer break and it will be of minimum 4 weeks. It will be evaluated at the end of VII Semester.

SEMESTER VII

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA401	Modeling & Simulation	3	1	0	4	CC3/ FC
2	CM401	Data Visualization	3	0	0	3	CC4/ SEC
3	CM403	Applied Machine Learning	2	0	0	2	GE4
4		Elective3	3	0	0	3	GE4
5		Elective4	3	0	0	3	GE6
6	CM481	Applied Machine Learning Lab	0	0	3	2	OE2/AECC
7	CM491	Minor project	0	0	10	5	CC-L2/SEC
8	CM493	Industrial Training	0	0	6	3	IT1/E
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**14**	**1**	**19**	**25**	

SEMESTER VIII

Sr.No	CourseCode	Courses	L-T-P	Credits
1	CS490	Seminar	0-0-3	2
2	CS492	Major Project	0-0-16	8
3	CS494	Internship	0-0-30	15
4	GP	General Proficiency	Non Credit	
		Total Credits		**25**
		Total Contact Hours	**0-0-49=25**	

ELECTIVES

S.No.	Course Code	CourseName	L	T	P	Credits	Types
1	CM311	Computational Intelligence	3	0	0	3	E1
2	CM313	Stochastic Process	3	0	0	3	E1
3	CM315	Machine Learning Algorithms	3	0	0	3	E1
4	CM317	Decision Thinking & Algorithm Design	3	0	0	3	E1
5	CM319	Statistical Machine Learning	3	0	0	3	E1
6	CM312	Artificial Neural Networks	3	0	0	3	E2
7	CM314	Knowledge Engineering	3	0	0	3	E2
8	CM316	Graph Theory	3	0	0	3	E2
9	CM318	Expert Systems	3	0	0	3	E2
10	CM320	Fuzzy logic	3	0	0	3	E2
11	CM405	Computational Neuroscience	3	0	0	3	E3
12	CM407	Intelligent Machining	3	0	0	3	E3
13	CM409	Introduction to Brain & Neuroscience	3	0	0	3	E3
14	CM411	Digital Fabrication	3	0	0	3	E3
15	CM413	Internet of things	3	0	0	3	E3
16	CM415	Digital Image Processing	3	0	0	3	E4
17	CM417	Ensemble Learning	3	0	0	3	E4
18	CM419	Predictive Analysis	3	0	0	3	E4
19	CM421	Embedded Systems	3	0	0	3	E4
20	CM423	Machine Intelligence for Medical Image	3	0	0	3	E4

GRAND TOTAL OF CREDITS = 200

- 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 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.

SEMESTER-1

FUNDAMENTALS OF COMPUTER PROGRAMMING			
Course Code:	**CS 101**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
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. To provide knowledge of primary and derived datatypes used in C
2. To make them understand basic conditional and break statements used in C
3. To provide a basic understanding of pointers and pointers arithmetic
4. To enable the students to explore how pre-defined functions are used and also created in a Program
5. Learn difference between static and dynamic memory allocation method and also learn various dynamic memory allocation methods.

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

UNITI INTRODUCTION TO COMPUTER AND PROGRAMMINGCONCEPTS

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.

UNITII INTRODUCTIONTOCPROGRAMMINGLANGUAGE

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.

UNITIII DATA TYPES ANDSTRUCTURES

bitwise operators, Pre defined and User defined data types, arrays, declaration and opera- tions 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, func- tions, declaration and use of functions, parameter passing, recurssion .

UNITIV FUNDAMENTALS OFPOINTERS

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

UNITV FILE HANDLING IN C ANDENUM

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 by HerbertShield

2. C Programming Language 2nd Edition by Brian, W Kernighan PearsonEducation.

3. Programming in ANSI C by E. Balagurusamy, Tata MgrawHill

4. C Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer PrenticeHallGale

5. Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling KindersleyIndia.

Fundamental of Machine Learning			
Course Code:	**CM101**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**02 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**30 + 00**	**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 practicalproblems.
3. To gain experience of doing independent study and research.
4. Learn the different type's machine learning techniques.
5. Summarize the Evaluation of Machine Learning
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
2. Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
3. Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
4. Be able to design and implement various machine learning algorithms in a range of real-world applications
5. Recognize the characteristics of machine learning that make it useful to real-world Problems

UNIT I INTRODUCTION TO MACHINELEARNING

History of Machine Learning, Introduction to Machine Learning, Applications of AI Well defined learning problems, Advantages of Machine Learning, challenges of machine learning, Applications of Machine Learning.

UNITII ARTIFICIAL NEURAL NETWORKS

Introduction to Biological Neuron, Neural network Architecture, Artificial Neural Network, Building blocks of ANN, Multilayer networks and Back propagation.

UNITIII LEARNING & COMPUTATIONAL LEARNINGTHEORY

Designing a Learning System, Programs vs. learning algorithms, supervised learning, unsupervised learning and Reinforcement Learning, Classification, Clustering, Advantage and Disadvantages of Supervised and unsupervised learnings.

UNITIV DECISION TREE LEARNING & GENETICALGORITHM

Introduction, Construction and representation of decision tree, strength and weaknesses of decision tree approach, Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning. Phases in genetic algorithm, an illustrative example.

UNIT IV Introduction to Deep Networks:

Introduction to deep feed forward networks, convolutional neural networks, stacking, striding and pooling, Natural Language Processing, Text Mining, Decision Trees, Training and Validation, Regression Trees, Hidden Markov Models.

Text Books:

1. Mitchell Tom, Machine Learning. McGraw Hill,1997.
2. EthemAlpaydin, Introduction to Machine Learning,PHI
3. Chris Bishop, Pattern Recognition and MachineLearning

Reference Books:

1. T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer 2011. (Available for download on the authors' web-page: http://statweb.stanford.edu/~tibs/ElemStatLearn/)
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.(Electronic copy available through the Bodleianlibrary.
3. Christopher M. Bishop. Pattern Recognition and Machine Learning, Springer2007.
4. S. Haykin. Neural networks and learning machines. Pearson2008.

COMPUTER PROGRAMMING LAB			
Course Code:	**CS181**	**Course Credits:**	**1**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Labs (Hrs/Week):	**02**	**Mid Sem. Exam Hours:**	
Total No. of Labs :	**10**	**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, analyzes, 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.
1. Develop confidence for self-education and ability for life-long learning needed for Computer language. 5.

LIST OF EXPERIMENTS:

1. Write a program for thefollowing:

a) To find the reverse of a givennumber.

b) Calculate factorial of a number usingrecursion.

2. Write a program to take marks of a student of 5 subjects as an input and print the grade. Also create the same program usingswitch.

   marks<40 = FAIL
   marks>=40 and <=59 =GOOD
   marks>=59 and <80 =EXCELLENT
   marks>=80 = OUTSTANDING

3. Write a program to compute the length of a string using WhileLoop.

4. Write a program to print the following pattern:-

a) *

   **

```
***
****
*****
```

b)
```
   *
  **
 * **  * * **
```

c)
```
0
12
   3 45
      6 7 89
```

5. Write a program to compute and display the product of twomatrices.
6. Write a program to illustrate the difference between call by value and call byreference.
7. Write a program to check whether a given string is palindrome ornot.
8. Create a structure called STUDENT having name, reg no., class as its field.

   Compute the size of structure STUDENT.
9. Write a program to compute the length of a string usingpointers.
10. Write a program to create a file, input data and display itscontent.

SEMESTER-2

Introduction to PYTHON			
Course Code:	**CM102**	**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 +0+0**	**Mid Sem. Exam Hours:**	**1**
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 BASICS, CONDITIONAL &LOOPS

Installation of Python and python Notebook, Python Objects, Number & Booleans, Strings, 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 comprehensions.

UNIT III TUPLES, SET, DICTIONARIES & FUNCTIONS

Tuples, Sets, Dictionary Object basics, Dictionary Object methods, Dictionary View Objects. Functions basics, Parameter passing, Iterators

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.

Text Books:

1. Head First Python 2e: A Brain-Friendly Guide Paperback – Illustrated, 16 by PaulBarry, Oreilly

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

3. Let Us Python by YashavantKanetkar , 1 January 2019, BPBpublication

4. Python Programming, A modular approach , First Edition, By Pearson Publication by Taneja Sheetal and Kumar Naveen , 26 September2017

COMPUTER ORGANIZATION AND ARCHITECTURE			
Course Code:	**CM104**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Discuss the basic concepts and structure of computers.
2. Understand concepts of register transfer logic and arithmetic operations.
3. Explain different types of addressing modes and memory organization.
4. Learn the different types of serial communication techniques.
5. Summarize the Instruction execution stages.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
2. Understand the theory and architecture of central processing unit
3. Analyze some of the design issues in terms of speed, technology, cost, performance.
4. Use appropriate tools to design verify and test the CPU architecture
5. Learn the concepts of parallel processing, pipelining and interprocessor communication.

UNIT I COMPUTER ARITHMETIC AND NUMBER SYSTEM

Functional units of digital system and their interconnections, Logic gates, Boolean algebra, combinational circuits, flip flops, sequential circuits, Number representation; number system, fixed and floating pointnumber representation, arithmetic algorithms (addition, subtraction, booth multiplication).

UNIT II REGISTER TRANSFER AND MICROOPERATION

Register transfer language, bus and memory transfers, bus architecture, bus arbitration, arithmetic logic, shift micro operation, arithmetic logic shift unit, design of fast address.

UNIT III PROCESSOR DESIGN

Processor organization: general register organization, stack organization, addressing mode, instruction format, data transfer & manipulations, program control, reduced instruction set computer.

UNIT IV INPUT-OUTPUT ORGANIZATION

I/O interface, synchronous and asynchronous data transfer, strobe, handshaking schemes, modes of transfer, interrupts & interrupt handling, direct memory access, I/O channels ,input-output processor.

UNIT V MEMORY ORGANIZATION

Memory hierarchy, main memory (RAM and ROM Chips), organization of 2d and 2 1/2 d, auxiliary memory, Cache memories: concept and design issues & performance, address mapping and replacement, virtual memory, memory management hardware.

Text Books:

. Patterson, Computer Organisation and Design, Elsevier Pub.2009
. William Stalling, " Computer Organization",PHI
. Vravice,Hamacher&Zaky, "Computer Organization",TMH
. Mano," Computer System Architecture",PHI
. John P Hays, " Computer Organization", McGrawHill
. Tannenbaum," Structured Computer Organization',PHI
. P Pal chaudhry, ' Computer Organization & Design',PHI

PYTHON PROGRAMMING LAB			
Course Code:	**CM182**	**Course Credits:**	**1**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**2U**
No. of Labs (Hrs/Week):	**2 hrs)**	**Mid Sem. Exam Hours:**	**1**
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To introduce students to use of Python programming to solve data analytics problems
2. To elaborate students to statistical analysis using Python programming
3. To describe various libraries required for data analytics
4. To elaborate statistical analysis using Python
5. To study special libraries in Python such as Numpy and Scipy

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Improve problem solving and programming capability
2. Learn data analytics through python programming
3. Underline the use of package
4. Write simple Python programs for solving problems.
5. Decompose a Python program into functions, lists etc.

List of Experiments:

Write a program in python:

1. To print the largest/smallest of twonumbers
2. To read two numbers x and n and print x^n (first write with the use of operator andthen write with the help of inbuiltfunction
3. To input the value of x and n and print the sum of theseries:

a. $1+x+x^2+x^3+x^4+\ldots\ldots x^n$

4. Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
5. Write a program to count the numbers of characters in the string and store them in a dictionary data structure
6. To print factorial of a number with and without using recursion
7. To tell the frequency of the most common word in a file or a given string
8. Write a function to find all duplicates in the list.
9. Write a program to perform addition and multiplication of two square matrices
10. To read from a text file and print each word separated by # symbol, example #xyz#xyz

SEMESTER 3

INTERNET TECHNOLOGY			
Course Code:	**CM201**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Present the basic web technology concepts for developing web applications.
2.Helps in computational thinking.
3.Understand of networking fundamentals.
4.Recognize the process of technology planning.
5.Interpret the paradigms of web page coding.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Have basic knowledge and understanding of core Internet technologies.
2.Apply Internet technology techniques for Web page design.
3.Learn various Browsing systems.
4.Work in JavaScript to create web pages effectively.
5.Process page Coding & Planning

UNITI OVERVIEW OF INTERNET ANDWEB

Introduction to internet, history of Internet and web, Internet services and accessibility, uses of internet, Internet standards, Internet protocols- IP, TCP,UDP and host names, web server, proxy server, fast ready connections on the web, web browsers, Netscape communication suite, Microsoft Internet explorer, firewalls, datasecurity.

UNIT II WEBDESIGN

Key issues in web site design, introduction to HTML, SGML- DTD, DTD elements, attributes, outline of an HTML document, body section- headers, paragraphs, text formatting, linking, internal linking, embedding images, lists, tables, frames, other special tags and characters, head section- prologue, link, base, meta, script, style, XML, XHTML, structuring data, XML schema documents, document object model, security and management issues for creating a website.

UNIT III BROWSING SYSTEMS

Searching and web casting technique, popular web servers, basic features, bookmarks, cookies, progress indicators, customization of browsers, browsing tricks, next generation web browsing, search engines, architecture of search engines, search tools, web crawlers, types of crawlers, scalable web crawler, incremental crawler, parallel crawler, focused crawler, agent based crawler, case study of IE, counters, Internet chat, hardware and software requirements for Internet and web based applications, Internet and web technologies.

UNIT IV JAVASCRIPT

Introduction,Languageelements,objectsofJavaScript,otherobjectslikedata,math,string,regularexpressions, andarrays.

UNIT V ACTIVE SERVER PAGES

Creatinginteractiveapplicationsusingactiveserverpages:clientandserversidescriptinC#,variablesand constants,creatingmodules,creatingobjectsfromclasses,ASP'sobjectmodel,arrays,collections,control structures, using request and response objects, Integration withdatabase.

Reference Books:

1. Raj Kamal, Internet and Web Technologies, TMH,2005.
2. MonicaD'Souza,Webpublishing,TMH,2001.
3. David Crowder and Rhonda Crowder, Web Design, IDG Books India,2001.
4. Musciano C., HTML and XHTML the Definitive Guide, 6th edition, OReilly,2006.
5. Deitel H.,Deitel P.,InternetandWorldWideWeb:HowtoProgram,4edition, PHI.

OPERATING SYSTEM			
Course Code:	**CM203**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**1U**	**Course Semester (U / P):**	**2U**
No. of Lectures + Tutorials (Hrs/Week):	**03+ 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):30	**45+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Understand how Operating System is Important for Computer System.
2.Make aware of different types of Operating System and their services.
3.Learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system
4.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:
1.Understand the different services provided by Operating System at different level
2.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.
4.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 OPERATINGSYSTEM
Importance of operating systems, basic concepts and terminology about operating system, memory management, processor management, device management, information management functions.

UNIT II PROCESSMANAGEMENT
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 &STORAGEMANAGEMENT
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, SystemAdministration.

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

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.

DATA STRUCTURE AND ALGORITHMS			
Course Code:	**CM205**	**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**
Total No. of Lectures (L + T):	**45 + 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, extendiblehashing.

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 linkedlist.

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 VGRAPHS

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. RobertL.Kruse,BruceP.LeungClovisL.Tondo,'DataStructuresandProgramDesigninC',Pearson Education, 2000 /PHI.

COMPUTER VISION			
Course Code:	**CM207**	**Course Credits:**	**3**
Course Category:		**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**
Total No. of Lectures (L + T):	**30 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

- To introduce students the majorideas.
- To Understand basic concepts ofCV
- To develop an appreciation for various issues in the design of computer vision and object recognitionsystem.
- Find and select appropriate data that can be used to create a visualization that answers a particular researchquestion.
- For each individual statistical test students should be able to understand how itworks

COURSE OUTCOMES

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

- identifybasicconcepts,terminology,theories,modelsandmethodsinthefieldofcomputervision
- describe known principles of human visualsystem
- describebasicmethodsofcomputervisionrelatedtomulti-scalerepresentation,edgedetection anddetectionofotherprimitives,stereo,motionandobjectrecognition,
- suggest a design of a computer visions system for a specificproblem

UNIT I Introduction to ComputerVision.

General introduction, History of CV, Required component, Useful application, Image acquisition using a camera, Different types of cameras for different domain- Stills, Video, DSLR, Bodycam, Drone, Color spaces: RGB, CMYK, HSV, Camera specifications: Pinhole, CMOS, CCD, Image specifications: Pixel (Picture element), Aspect ratio, HD, Interlacing, Type of digital images: Binary, Grayscale, Color, Conversion techniques.

UNIT II Image processing and Edge Detection

Noise Removal, salt and pepper noise, Pixel Neighborhood, Types of Filter: mean or Box filtering, median Filter, Generic properties of smoothing, Gaussian separability, Introduction to edges and gradient, Intensity difference, 1D versus 2D edge detection, Edge detection in mammals, 1D signals and 2D signals, Image Gradient, Image noise: Gaussian noise, Smoothing + Edge detection.

UNIT III Image Segmentation and features.

Image Segmentation and features, Thresholding based on histogram, formulation, Advancements, and effectiveness, Thresholding based on different metrics, covariance-based, Different types of background subtraction, mean, Euclidean, Mahalanobis, Clustering to Image Segmentation, Transform to color space.

UNIT IV Shape of Objects and Motion.

Medial axis, Boundary coding, Chain Coding, Shape Numbering, Bounding box, Principal Component Analysis, Eigen Values and Vectors, Finding Eigen sets, Simple motion, Image differentiation, Single constant threshold, Weighted aggregate, Hierarchical Motion Estimation, 3D motion of a point, Matrix operations for different motion in objects, 2D matrix motion, Translation Motion, Affine Motion, Spatial Pattern of where motion occurred.

UNIT V Feature Extraction and Camera Projection:

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, Ambiguity in single, Geometry for simple stereo system View, depth and Calibration, Epipolar Geometry: Baseline, Epipole, Epipolar Line, Epipolar Plane.

Text and Reference Books:

[1] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited2011.

[2] Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education,2003.

[3] Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March2004.

INTRODUCTION TO R PROGRAMMING			
Course Code:	**CM209**	**Course Credits:**	**3**
Course Category:		**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**
Total No. of Lectures (L + T):	**30 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Master the use of the R and RStudio interactive environment.
2.Expand R by installing R packages.
3.Explore and understand how to use the R documentation.
4.Understand the different data types in R.
5.Understand the different data structures in R.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.The course would enable the ability to understand and critically assess available data using machine learningmethods.
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.The course would also inform use of large volume data by extracting useful information and patterns and provide predictive insights.

UNITI INTRODUCTION TO R ANDRSTUDIO

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.

UNITII DATA STRUCTURES: VECTORS, MATRICES, LISTS AND DATAFRAMES

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.

UNITIIIREADING 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.

UNITIV STATISTICAL MODELINGFUNCTIONS

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

UNITV WRITING YOUR OWNFUNCTIONS

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 asarguments.

Text Books:

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

R PROGRAMMING LAB			
Course Code:	**CM281**	**Course Credits:**	**3**
Course Category:		**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**
Total No. of Lectures (L + T):	**30 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Install and set up R and RStudio.
2. Understand R data types
3. Understand R data structures
4. Understand R functions
5. Understand R Markdown
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Access online resources for R and import new function packages into the R workspace
2.Import, review, manipulate and summarize datasets in R
3.learn the main R data structures – vector and data frame
4.Explore datasets to create testable hypotheses and identify appropriate statistical tests.
5.Perform appropriate statistical tests using R.

List of Experiments:

1. WAP to read data from various sources in a dataframe.
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 to91.
3. Write a R program to find the factors of a givennumber.
4. Write a R program to find the maximum and the minimum value of a givenvector.
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 5 matrix with labels and fill the matrix bycolumns.
6. Write a R program to get the statistical summary and nature of the data ofa given dataframe.
7. Write a R program to create inner, outer, left, right join(merge) from given two data frames.Write a R program to save the information of a data frame in a file and display the information of the file.
8. Build a simple web app usingShiny.

9. 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 thelist.
10. Load the data set and create a dashboard usingshiny.
11. Connect the R script to dummy database and retrieve data from it and save it in a dataframe.

DATA STRUCTURE AND ALGORITHMS LAB			
Course Code:	**CM283**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Labs (Hrs/Week):	**2(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**
LAB 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.			
LAB OUTCOMES			
At the end of the course the students should be able to:			
1. Select appropriate data structures as applied to specified problem definition			
2.Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.			
3.Students will be able to implement Linear and Non-Linear data structures.			
4. Implement appropriate sorting/searching technique for given problem.			
5. Design advance data structure using Non-Linear data structure			

List of Experiments:

1. Run time analysis of FibonacciSeries
2. Study and Application of various dataStructure
3. Study and Implementation of Array BasedProgram
   a. Searching (Linear Search, BinarySearch)
   b. Sorting (Bubble, Insertion, Selection, Quick, Mergeetc)
   c. Merging
4. Implementation of LinkList
   a. Creation of Singly link list, Doubly Linkedlist
   b. Concatenation of Linklist
   c. Insertion and Deletion of node in linklist
   d. Splitting the link list into two linklist
5. Implementation of STACK and QUEUE with the helpof
   a. Array
   b. LinkList
6. Implementation of BinaryTree

7. Implementation of Binary SearchTree.

8. Write a program to simulate various traversingTechnique

   a. Representation and Implementation of Graph.Depth FirstSearch
   b. Breadth FirstSearch
   c. PrimsAlgorithm
   d. Kruskal'sAlgorithms

9. Implementation of HashTable

INTERNET TECHNOLOGY LAB			
Course Code:	**CM285**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**2U**
No. of Labs	**1(3 hrs)**		
Total No. of Lab(L + T):10	**10+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.To design interactive web pages using Scripting languages.
2.To learn server side programming using servlets and JSP.
3.To develop web pages usingXML/XSLT
4.To develop dynamic web pages using different platforms
5.Learn how to use XAMP Server
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Design simple web pages using markup languages like HTML and XHTML.
2.Create dynamic web pages using DHTML and java script that is easy to navigate and use.
3.Program server side web pages that have to process request from client side web pages.
4.Represent web data using XML and develop web pages using JSP.
5.Understand various web services and how these web services interact.

List of Programs

1. Create a web page with the following usingHTML.
   0. To embed an image map in a webpage.
   1. To fix the hotspots.
   2. Show all the related information when the hot spots areclicked
2. Create a web page with all types of Cascading stylesheets.
3. Client Side Scripts for Validating Web Form Controls usingDHTML.
4. Installation of Apache Tomcat webserver.
5. Write programs in Java usingServlets:
   0. To invoke servlets from HTMLforms.
   1. SessionTracking.
6. Write programs in Java to create three-tier applications using JSP andDatabases
   0. For conducting on-lineexamination.
   1. Fordisplayingstudentmarklist.Assumethatstudentinformationisavailableinadatabase which has been stored in a databaseserver.
7. Programs Using Xml – Schema –Xslt/Xsl.
8. Programs using DOM and SAXparsers.
9. Programs usingAJAX.

10. Consideracasewherewehavetwowebservices-anairlineserviceandatravel agentandthetravel agent is searching for an airline. Implement this scenario using Web Services and Database.

Software Required:

- Dream Weaver or Equivalent, MySQL or Equivalent, Apache Server,WAMP/XAMPP

SEMESTER-4

SOFTWARE ENGINEERING			
Course Code:	**CM202**	**Course Credits:**	**3**
Course Category:CC	**CC**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**4 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Help students to develop skills that will enable them to construct software of high quality software that is reliable, and that is reasonably easy to understand, modify and maintain.
2. Foster an understanding of why these skills are important
3.Provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects
4.Study process models, software requirements, software design, software testing
5.Help to study Software process/product metrics, risk management, quality management and UML diagrams

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
2. Expertise and/or awareness of testing problems and will be able to develop a simple testing report
3.Translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
4. Analyse various software engineering models and apply methods for design and development of software projects
5 .Proficiently apply standards, CASE tools and techniques for engineering software projects

UNIT I SOFTWARE ENGINEER

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, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

UNIT II REQUIREMENT ANALYSIS ANDDESIGN

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, Decision Tables.

Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality

Frameworks, ISO 9000 Models, SEI-CMM Model.

UNIT III SOFTWARE DESIGNPROCESS

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. Software Measurement and Metrics: Various Size OrientedMeasures:Halestead'sSoftwareScience,FunctionPoint(FP)BasedMeasures,CyclomaticComplexity Measures: Control FlowGraphs.

UNIT IV SOFTWARETESTING

Testing Objectives, Unit Testing, Integration Testing, 8 Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products.Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

UNIT V SOFTWAREMAINTENANCE

Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.problem resolution, software maintenance from customers" perspective, maintenance standard: IEEE-1219, ISO-12207, Software Risk Analysis and Management.

Text Books:

1. PankajJalote,AnIntegratedApproachtoSoftwareEngineering,NarosaPublishingHouse,NewDelhi 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.

DATABASE MANAGEMENT SYSTEM			
Course Code:	**CM204**	**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 the fundamental elements of relational database management systems
2.Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
3.Design ER-models to represent simple database application scenarios
4.Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.
5.Improve the database design by normalization.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand of database concepts and thorough knowledge of database software's.
2.Model an application's data requirements using ER diagrams
3. Write SQL commands to create tables and query data in a relational DBMS
4. Execute various advanced SQL queries related to transactions, concurrency
5.Explain the principle of transaction management design.

UNIT I DATA BASESYSTEM

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 RELATIONALMODEL

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 SQLQUERY

Examples of basic SQL queries, nested queries, correlated nested queries set, comparison operators,
aggregativeoperators,NULLvalues,comparisonusingnullvalues,logicalconnectivity's,AND,ORandNOTR, impact on SQL constructs, outer joins, disallowing NULL values, complex integrity constraints in SQL triggers and active databases.

UNIT IV SCHEMAREFINEMENT

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 TRANSACTIONMANAGEMENT

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, DatabaseSystemConcepts,McGrawhill,5thedition,2005.
5. Rob,Coronel &Thomson,DatabaseSystemsDesign:ImplementationandManagement,2009.

JAVA PROGRAMMING			
Course Code:	**CM206**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Teach principles of object-oriented programming paradigm including abstraction, encapsulation, inheritance, and polymorphism.
2.Impart fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
3.Familiarize the concepts of packages and interfaces
4.Facilitate students in handling exceptions.
5.Demonstrate the concept of event handling used in GUI.
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Analyze the necessity for Object Oriented Programming paradigm over structured programming and become familiar with the fundamental concepts in OOP like encapsulation, Inheritance and Polymorphism
2.Design and develop java programs, analyze, and interpret object-oriented data and report results
3.Design an object-oriented system, AWT components and multithreaded processes as per needs and specifications.
4.Participate and succeed in competitive examinations like GATE, Engineering services, recruitment interviews etc.
5.Plan their career in java-based technologies like HADOOP etc.

UNIT I OBJECT-ORIENTED PROGRAMMING

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history, Java features, Java streaming, Java and Internet, Java contribution to Internet: Java applets, security, portability; Java environment, Java library, Java program structure, Java program, Java Virtual Machine (JVM) architecture, Just In Time compiler (JIT), data type, variables and arrays, operators, control statements, object-oriented paradigms; abstraction, encapsulation, inheritance, polymorphism, Java class and OOPimplementation

UNIT II DATA TYPE, OPERATORS AND CONTROLSTATEMENT

Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable, symbolic constant, type casting, arithmetic operator, relational operator, logical operator, assignment operator, increment and decrement operator, conditional operator, bitwise operator, ?: operator, arithmetic expressions, expressions, type conversions in expressions, mathematical functions, more data types: arrays, strings, vectors, wrappersclasses,programcontrolstatements:decisionmakingandbranching:if,if….else,else….if,elseif ladder,switch,decisionmakingandlooping:while,do….while,for.

UNIT III CLASSES, OBJECTS ANDMETHODS

Java class libraries, class fundamentals, object, methods, adding variables, add methods, creating objects, accessing class members, constructors, methods overloading, static members, nesting of methods, inheritance: extending a class, overriding methods, final variables and methods, final classes, finalizer methods, abstract methods and classes, visibility control, exception handling fundamental.

UNIT IV INTERFACES ANDPACKAGES

Interfaces, extending interfaces, implementing interfaces, interfaces references, accessing interface variable, creating queue interface, variable in interfaces, packages, finding a packages and classpath, package and member access, Java API package, system package, naming conventions, creating package, accessing a package, adding a class to a package, hiding classes.

UNIT V MULTITHREADING AND APPLETPROGRAMMING

Multithreading programming: creating threads, thread class and runnable interface extending the thread class, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, thread priority, synchronization, thread communication using notify(), wait(), and notify all(), applet programming : applet basic, applets architecture, a complete applet skeleton, building applets code, applets life cycle, creating a executable applet, designing a web page, applets tag, passing parameters to applets, applets and HTML.

Text Books:

1. Programming with JAVA, E. Balagurusawamy, Tata McGraw Hill,1998.
2. JAVABeginner‟sguide,HerbertSchildt,TataMcGrawHill,2007.
3. Java How to Program, Deitel&Deitel, Prentice-Hall,1999.

ARTIFICIAL INTELLIGENCE			
Course Code:	**CM208**	**Course Credits:**	**2**
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**
Total No. of Lectures (L + T):	**45 + 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 of Artificial 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 and learning			
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 asystem.			
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, Advantages and Disadvantages of AI, Intelligence, Intelligent System, Role of IS, Comparison of various IS, Weak AI 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 AI Agents, Simple Reflex Agent,Model-based reflex agent, Goal-based agents, Utility- based agent, Learning agent, Structure of an AI Agent, Agent Environment in AI, Examples of Agents, Knowledge Engineering, Knowledge Based System, Knowledge Engineering Techniques, Knowledge Engineering Principles, Knowledge Engineering Methodology.

UNIT 3 SEARCHING TECHNIQUES AND AI PROBLEMS

Searching in AI, Search Algorithm Terminologies, Properties of Search Algorithms, Breadth-first search, Depth- first search, Best First Search, Tic-Tac Toe Problem, Water Jug problem, Chess Problem, Tower of Hanoi problem, Travelling Salesman problem, 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 knowledgecycle.

UNIT 5 AI Techniques and applications

Introduction to Machine Learning, Introduction to Deep Learning, Introduction to Expert system, Introduction to Natural Language Processing, AI in future, AI in social Media, AI in Entertainment and education, AI in drones, AI in Automated Computer support, AI in personalized shopping experience, AI in Finance, AI in smart Cars, AI in travel and navigation, AI in smart home devices, AI in security and surveillance, Ai in education, AI in health care, AI in Ecommerce.

ReferenceBooks:

1. Artificial Intelligence, Elaine Reich: Tata Mcgraw Hill publishing house,2008.
2. Artificial Intelligence, Ela Kumar, IKPublishing.
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 & PeterNorvig.
7. Artificial Intelligence: The Basics By KevinWarwick

THEORY OF AUTOMATA			
Course Code:	**CM210**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Determine the various categories of automata (deterministic and nondeterministic finite state automata, and variants of Turing machines)			
2.Understand the various categories of languages and grammars in the Chomsky hierarchy			
3.Define the notions of computability and decidability			
4.Recognize to which class in the Chomsky hierarchy the language described (by a grammar or machine)			
5. Discover the problems reducible to/from well-known decidable/undecidable problems			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1.Model, compare and analyse different computational models using combinatorial methods.			
2.Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.			
3.Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.			
4.Identify limitations of some computational models and possible methods of proving them.			
5.Have an overview of how the theoretical study in this course is applicable to and engineering application like designing the compilers.			

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, Myhill-Nerode Theorem.

UNIT II REGULAR EXPRESSION

Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic Laws for Regularexpressions,Kleen'sTheorem,RegularexpressiontoFA,DFAtoRegularexpression,ArdenTheorem, 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 IV PDA

Push Down Automata (PDA): Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

UNIT V Turing machines (TM)

Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church's Thesis, Recursive and recursively enumerable languages, Halting 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.MishraandN.Chandrasekaran,"TheoryofComputerScience: Automata, Languagesand Computation",PHILearningPrivateLimited,DelhiIndia

INTRODUCTION TO MATLAB			
Course Code:	**CM 212**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Create and troubleshoot basic m scripts.			
2. Create publishable, reproducible analysis reports.			
3. Confidently develop MATLAB M-files and save results of computations from a MATLAB session.			
4. . Use MATLAB to perform complex arithmetic			
5.Generate and plot signals and complex valued functions			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Utilize a methodical approach to identify, formulate, and solve computational problems.			
2. Comprehend MATLAB basics, branching and looping.			
3. Apply MATLAB in solving algebra calculus problems.			
4. Apply various techniques to solve and visualize engineering-related computational problems using MATLAB.			
5. Sketching of discrete and continuous time signals			

UNIT I: Introduction and Basics.

Importance of MATLAB, MATLAB environment, various toolboxes, using MATLAB as a calculator, Variables and arrays, operations on variables and arrays, matrix operations, displaying output data, introduction to plotting, data files, built-in MATLAB functions, user-defined function.

UNIT 2: Branching and Loops and 2D/3D Plots.

Relational and logic operators, branches, WHILE loops FOR loops, SWITCH, BREAK, CONTINUE, vectorization, MATLAB profiler, 2D plots, 3D plots, data distribution plots, polar plots, contour plots, surface plots .

UNIT 3: Numerical Methods and Structures.

Linear algebra and vector analysis, newton and bisection methods, numerical solution to ordinary differential equations, curve fitting, interpolation, least squares regression , Cell Arrays, structure arrays, string, sorting & searching, importing data into MATLAB, file Input / Output functions, working with the spreadsheet and low- level data file.

UNIT 4: Advanced Features

Graphical User Interfaces and GUIDE, application development, Simulink, MATLAB with crosslanguage platforms

UNIT 5: Applications.

Optimization methods, Signal processing, image processing, machine learning, system-level modeling.

Text Books:

1.Matlab for Beginners: A Gentle Approach Peter I. Kattan2008

2. Matlab for Newbies: The bare essentials, September 9, 2015,by Siddharth Verma.

3. MATLABHandbookwithApplicationstoMathematics,Science,Engineering,andFinance Jose

Miguel David Baez-Lopez, David Alfredo Baez Villegas2019

DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**CM282**	**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(3 hrs)**		
Total No. of Labs	**10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Explain basic database concepts, applications, data models, schemas and instances.			
2.Demonstrate the use of constraints and relational algebra operations.			
3.Emphasize the importance of normalization in databases.			
4.Facilitate students in Database design			
5.Familiarize issues of concurrency control and transaction management.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1.Students get practical knowledge on designing and creating relational database systems.			
2.Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger, views and embedded SQL.			
3.Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system			
4.Use the basics of SQL and construct queries using SQL in database creation and interaction.			
5.Analyze and Select storage and recovery techniques of database system.			

List of Experiments:

1. Introduction to MySQL, an exercise of data types in MySQL &Data Definition LanguageCommands
2. Exercise on Data Manipulation Language and Transaction ControlCommands
3. Exercise on Types of DataConstraints
4. Exercise on JOINS (Single-Table) UsingNormalization
5. Exercise on JOINS (Multiple-Table) UsingNormalization
6. Exercise on GROUP BY/ORDER BY Clause and DateArithmetic
7. Exercise on different Functions (Aggregate, Math andString)
8. Exercise on different types of subqueries
9. Procedures

JAVA PROGAMMING LAB			
Course Code:	**CM284**	**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):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Prepare students to excel in Object Oriented programming and to succeed as a Java Developer through global rigorous education
2. Students learn an object-oriented way of solving problems using java.
3. Make the students to write programs using multithreading concepts and handle exceptions.
4.Demonstrate the students to write programs that connects to a database and be able to perform various operations.
5.Make the students to create the Graphical User Interface using Applets, AWT Components &Swing Components.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.To Understand OOP concepts and basics of Java programming.
2.Design and develop java programs, analyze, and interpret object-oriented data and report results.
3.Demonstrate an ability to design an object-oriented system, AWT components or multithreaded process as per needs and specifications.
4.To build files and establish database connection.
5.To visualize and work on laboratory and multidisciplinary tasks like console and windows applications both for standalone and Applets programs

1. Write a separate Java Code to implement each of thefollowing:

   Class, Command Line Argument, how to enter value through keyboard
2. WriteaseparateJavaCodetoimplementeachofthefollowingdatatypes:Variable,Constant, Arrays, Strings, Vectors, Wrappers Classes, TypeCasting

3. Write a separate Java Code to implement each of the followingoperators:

   Arithmetic operator, Relational operator, Logical operator, Assignment operator, Increment & Decrement operator, Conditional operator, Bitwise operator, ?: operator
4. WriteaseparateJavaCodetoimplementeachof thefollowingcontrolstatements:Decisionstatement, Loops statement and Branchstatements

5. WriteaseparateJavaCodetoimplementeachofthefollowingsorting:BubbleSort,SelectionSort, Insertion Sort, MergeSort

6. Write a separate Java Code to implement each of thefollowing:

   Class, Object, Constructors, Method, Method Overloading and Method Overriding
7. Write a separate Java Code to implement each of thefollowing:

   Final variable, final class, final method, abstract class, abstract method and concrete method
8. Write a separate Java Code to implement each of the following OOPs concepts: Abstraction, Polymorphism, Encapsulation,Inheritance

9. Write a separate Java Code to implement each of the following: Exception handling with Try, Catch, Throw, Throws, Finally Multiple catch statement with the following exceptions : ArithmeticException, ArrayOutOfBoundsException andArrayStoreException

10. Write a separate Java Code to implement thefollowing:

a) Interface

b) Packages and how to importthem.

MATLAB			
Course Code:	**CM286**	**Course Credits:**	**2**
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:**	
Total No. of Lectures (L + T):	**10+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 To Impart the Knowledge to the students with MATLAB software.
2 To provide a working introduction to the Matlab technical computing environment..
3 To introduce students the use of a high-level programming language, Matlab..
4 Being able to do simple calculations using MATLAB.
5 Being able to carry out simple numerical computations and analyses using MATLAB.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand the basics of Matlab.
2 Break a complex task up into smaller, simpler tasks.
3. Manipulate vectors and matrices, use matrix indexing, and determine matrix dimensions
4 Write simple programs in MATLAB to solve scientific and mathematical problems.
5 Use the MATLAB GUI effectively.

LIST OF EXPERIMENTS:

1. Introduction to SDK ofMATLAB.
2. Basic Syntax and scalar arithmetic operations andcalculations.
3. Working withformulas.
4. Arithmetic operations in matrixdata
5. Matrix operations (Inverse, Transpose)
6. Reading an imagefile
7. Reading from and writing to a textfile
8. Introduction totoolboxes
9. Data visualization andplotting
10. Relational operators indata
11. Logical operation indata
12. LoopsinMATLAB
13. Computing Eigen value for amatrix
14. Random number generation – Montecarlomethods

SEMESTER-5

COMPILER DESIGN			
Course Code:	**CM301**	**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.Find Out the relations between computer architecture and how its understanding is useful in design of a compiler.
3. 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.Analysethetargetmachine'sruntimeenvironment,itsinstructionsetforcodegenerationand techniques used for code optimization.

UNIT I INTRODUCTION TOCOMPILER

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, syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

UNIT II PARSINGTECHNIQUE

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 parsingtables.

UNIT III SYNTAX-DIRECTEDTRANSLATION

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 SYMBOLTABLES

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 CODEGENERATION

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. Aho,Sethi&Ullman,"Compilers:Principles,TechniquesandTools",PearsonEducation

2. VRaghvan,"PrinciplesofCompilerDesign",TMH

3. KennethLouden,"CompilerConstruction",CengageLearning.

SOFT COMPUTING TECHNIQUES			
Course Code:	**CM303**	**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**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of soft computing.
2. Understanding of the basic areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.
3.Provide the mathematical background for carrying out the optimization associated with neural network learning.
4.Aim of this course is to develop some familiarity with current research problems and research methods in Soft Computing by working on a research or design project.
5. Genetic algorithms, its applications and advances.
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Apply basics of Fuzzy logic and neural networks..
2.Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human
3. Describe with genetic algorithms and other random search procedures useful while seeking global optimum in self-learning situations
4. Develop some familiarity with current research problems and research methods in Soft Computing Techniques
5. experience Relate with neural networks that can learn from available examples and generalize to form appropriate rules for inference systems

UNITI INTRODUCTION

Introduction to Soft Computing; Definition, requirement, necessity and adequacy; various dialects of soft computing – Evolutionary Algorithms, Fuzzy Sets and Fuzzy Logic, Artificial Neural Networks - their suitability in Searching, optimization, decision matching and pattern related problems; potential areas of applications.

UNITII FUZZY SETS AND FUZZYLOGIC

Introduction to fuzzy sets and fuzzy logic; difference between classical and fuzzy sets; chance vs fuzziness; limitations of fuzzy systems; typical shapes of membership functions and their usage; operations on fuzzy sets: compliment, intersection, union; combinations on operations, aggregation operation.

UNITIII FUZZY RELATIONS AND FUZZYSYSTEMS

Cartesian Product; Classical Relations and Fuzzy Relations; Cardinality, operations and properties of crisp and fuzzy relations; Composition of operations, Fuzzy cartesian product; The linguistic variables, Reasoning in fuzzy logic, Fuzzification and defuzzification; Mamdani and Sugano Fuzzy Inference Systems.

UNITIV NEURALNETWORK

Overview of biological neurons; McCulloch-Pitts model, Rosenblatt's Perceptron model, difference, capabilitiesand limitations; Model of generic computational neuron; Basic activation functions; Basic Learning laws of neurons; Single layer and multilayer architectures; Feedforward and feedback networks.

UNITV LEARNINGFUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning; back propagation algorithm; Radial basis neurons, Generalized Regression Neural network, Probabilistic Neural Networks; Competitive learning; Self Organizing Features Map, Hopfield networks, associative memories, applications of artificial neural networks. Elasticity vs plasticity dilemma, preprocessing, post processing, early stopping.

UNITVI EVOLUTIONARYALGORITHMS

Problems suitable and not suitable for applying evolutionary algorithms; Various dialects of evolutionary Algorithms; Terminology of Genetic Algorithms; Canonical Genetic Algorithm; Common representations and related reproduction operators; premature convergence, schema theorem, minimal deceptive problem and Royal Road function; fitness function, Roulette wheel selection, Rank selection, Tournament Selection; termination criteria, survivor selection, population models; parallel implementations.

Text Books:

1. Artificial Neural Networks: An introduction to ANN Theory and Practice, Peteus J.Braspenning,

   PHI publication, 2005.
2. Fuzzy Logic: A spectrum of Theoretical and Practical issues, Paul P. Wang, pearson publication2004.
3. An Introduction to Genetic Algorithms, Milanie Mitchell, MIT Press1998.
4. A Genetic Algorithm Tutorial, DarrellWhitley.
5. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J. Kilr, Bo yuan,2005.
6. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conference proceeding, 2005.
7. Neural Networks Theory, ParticiaMelin, Oxford University press,2003
8. Neural Networks Theory and Application, Oscar Castillo, Wiley Easternpublication
9. Genetic Algorithms in Search, Optimization and Machine Learning, David E Goldberg, Eddison-Wesley, 1988.

ANALYSIS & DESIGN OF ALGORITHMS			
Course Code:	**CM305**	**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**
Total No. of Lectures (L + T):	**45 + 15**	**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.Define 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

References Books:

1. T.H. Cormen, C.E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithm", PHI Pvt. Ltd., 2001

2. Sara Baase and Allen Van Gelder,"Computer Algorithms-Introduction to the Design and Analysis ", Pearson Education Asia,2003

3. A. V. Aho, J.E. Hopcroft and J.D. Ullman, "the Design and Analysis of Computer Algorithms", Pearson Education Asia,2003.

BIG DATA ANALYTICS			
Course Code:	**CM307**	**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. 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. Explain the Map Reduce Jobs
5. Apply analytics on Structured, Unstructured Data. Exposure to Data Analytics with R.
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Identify Big Data and its Business ImplicationsAccess and Process Data on Distributed File System
2.List the components of Hadoop and Hadoop Eco-System
3.Manage Job Execution in Hadoop Environment
4. Develop Big Data Solutions using Hadoop Eco System
5.Analyze Infosphere BigInsights Big Data Recommendations

UNITI INTRODUCTION

Introduction to big data : Introduction to Big Data Platform, Challenges of Conventional Systems, Intelligent data analysis, Nature of Data, Analytic Processes and Tools, Analysis vs Reporting.

UNIT II DATA STREAMS

Mining data streams : Introduction To Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream , Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform(RTAP) Applications - Case Studies - Real Time Sentiment Analysis- Stock Market Predictions.

UNIT III HADOOP

Hadoop: History of Hadoop, the Hadoop Distributed File System, Components of Hadoop Analysing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfaces to HDFS Basics, Developing a Map Reduce Application-How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling- Shuffle and Sort – Task execution, Map Reduce Types and Formats- Map Reduce FeaturesHadoop environment.

UNIT IV DATAPROCESSING

Frameworks: Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services, HiveQL,QueryingDatainHive,fundamentalsofHBaseandZooKeeper,IBMInfoSphereBigInsightsandStreams.

UNIT V DATA ANALYTICS TECHNIQUE

Predictive Analytics- Simple linear regression, Multiple linear regression., Interpretation 5 of

regression coefficients. Visualizations, Visual data analysis techniques, interaction techniques, Systems and applications.

Text Books:

1. MichaelBerthold,DavidJ.Hand,"IntelligentDataAnalysis",Springer,2007.
2. TomWhite"Hadoop:TheDefinitiveGuide"ThirdEdition,O'reillyMedia,2012.
3. ChrisEaton,DirkDeRoos,TomDeutsch,GeorgeLapis,PaulZikopoulos,"UnderstandingBigData

MACHINE LEARNING			
Course Code:	**CM309**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
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. Explain Machine Learning concepts, classifications of Machine Learning and write simple programs using python.
2.Describe Supervised Learning concepts.
3.Describe unsupervised learning concepts and dimensionality reduction techniques
4.Discuss simple Machine Learning applications in a range of real-world applications using Python programming
5.To develop skills of using recent machine learning software for solving practical problems.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Recognize the characteristics of machine learning that make it useful to real-world problems.
2.Characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
3.Effectively use machine learning toolboxes.
4.Understand the concept behind neural networks for learning non-linear functions.
5.Figure out the algorithms for learning Bayesian networks

Unit 1: Introduction – Well defined learning problems, Designing a Learning System, Issues in Machine Learning; THE CONCEPT LEARNING TASK - General-to-specific orderingof hypotheses, Find-S, List then eliminate algorithm, Candidate elimination algorithm, Inductive bias

Unit 2: Decision Tree Learning - Decision tree learning algorithm-Inductive bias- Issues in Decision tree learning; ARTIFICIAL NEURAL NETWORKS – Perceptrons, Gradient descent and the Delta rule, Adaline,Multilayer networks, Derivation of backpropagation rule Backpropagation Algorithm Convergence, Generalization

Unit 3: Evaluating Hypotheses: Estimating Hypotheses Accuracy, Basics of sampling Theory, Comparing Learning Algorithms;

Bayesian Learning: Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm;

Unit 4: Computational Learning Theory: Sample Complexity for Finite Hypothesis spaces, Sample Complexity for Infinite Hypothesis spaces, The Mistake Bound Model of Learning;

INSTANCE-BASEDLEARNING–k-NearestNeighbourLearning,LocallyWeightedRegression, Radial basis function networks, Case-basedlearning

Unit 5: Genetic Algorithms: an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning; Learning first order rules-sequential covering algorithms- General to specific beam search-FOIL; REINFORCEMENT LEARNING - The Learning Task, Q
Learning.

Text Books:

1.TomM.Mitchell,―MachineLearning,McGraw-HillEducation(India)PrivateLimited,2013.
2. EthemAlpaydin,―IntroductiontoMachineLearning(Adaptive Computationand Machine Learning), The MIT Press 2004.
3. StephenMarsland,―MachineLearning:AnAlgorithmicPerspective,CRCPress,2009.Bishop, C., Pattern Recognition and Machine Learning. Berlin:Springer-Verlag.

ANALYSIS & DESIGN OF ALGORITHMS LAB			
Course Code:	**CM381**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):U	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**02(3 hrs)**		
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-Conquertechnique
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.ExperimentfindingtheminimumcostofspanningtreeusingPrim'salgorithmsandshortestpath using Dijkstra' algorithm

PRACTICALS

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

1.Sort a given set of elements :

(a)usingtheQuicksortmethodandalsoanalyseit'sruntimecomplexityfordifferentinputs.

(b)usingmergesortmethodandalsoanalyseit'sruntimecomplexityfordifferent inputs.

2. Write a program to obtain the topological ordering of vertices in a givendigraph.

3. Implement travelling salesmanproblem.

4. Implement the knapsack problem(0/1).

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

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

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

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

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

10.From agivenvertexinaweightedconnectedgraph,findshortestpathstootherverticesusingDijkstra's algorithm.

BIG DATA ANALYTICSLAB			
Course Code:	**CM383**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):U	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.Understand the Big Data Platform and its Use cases
2. Provide an overview of Apache Hadoop & exposure to Data Analytics with R.
3. Provide HDFS Concepts and Interfacing withHDFS
4. Understand Map Reduce Jobs & apply analytics on Structured, Unstructured Data
5. Provide hands on Hodoop Eco System

COURSE OUTCOMES
1.Implement numerical and statistical analysis on various data sources.
2. Apply data preprocessing and dimensionality reduction methods on raw data
3.Implement linear regression technique on numeric data for prediction .
4. Execute clustering and association rule mining algorithms on different datasets
5.Implement and evaluate the performance of KNN algorithm on different datasets

PRACTICALS List.

1. PerformsettingupandInstallingHadoopinitsthreeoperatingmodes:Standalone,Pseudodistributed, Fully distributed.
2. Implement the following file management tasks inHadoop:
3. Adding files anddirectories
4. Retrievingfiles
5. 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 lineutilities.
6. Run a basic Word Count Map Reduce program to understand Map ReduceParadigm.
7. WriteaMapReduceprogramthatminesweatherdata.Weathersensorscollectingdataeveryhouratmany locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured andrecord-oriented.
8. Implement Matrix Multiplication with Hadoop MapReduce
9. InstallandRunPigthenwritePigLatinscriptstosort,group, join,project,andfilteryourdata.
10. Install andRunHivethenuseHivetocreate,alter,anddropdatabases,tables,views,functions,andindexes.
11. Solve some real life big dataproblems.
12. To perform market basket analysis using Association Rules(Apriori).
13. To perform dimensionality reduction operation using PCA for Houses DataSe

MACHINE LEARNING LAB USING PYTHON			
Course Code:	**CM385**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Labs (Hrs/Week):	**2(3 hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.To understand the basic concepts and techniques of Machine Learning through python programming.
2.To develop skills of using recent machine learning packages for solving practical problems.
3.To gain experience of doing independent study and research
4.To understand the methods using in machine learning
5. To demonstrate real time applications using python

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Familiarize Python
2.Able to generate, analyze and interpret data using Python.
3. Use Python to design and implement classifiers for machine learning applications.
4.Implement an end to end Machine Learning System
5. Design new programs related to machine learning methods

List of Experiments:

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, StandardDeviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy andScipy
3. Study of Python Libraries for ML application such as Pandas andMatplotlib
4. Write a Python program to implement Simple LinearRegression
5. Implementation of Multiple Linear Regression for House Price Prediction usingsklearn
6. Implementation of Decision tree using sklearn and its parametertuning
7. Implementation of KNN usingsklearn
8. Implementation of Logistic Regression usingsklearn
9. Implementation of K-MeansClustering
10. Performance analysis of Classification Algorithms on a specific dataset (MiniProject)

ELECTIVE I

COMPUTATIONAL INTELLIGENCE			
Course Code:	**CM311**	**Course Credits:**	**3**
Course Category:	**E1**	**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.

STOCHASTIC PROCESS			
Course Code:	**CM313**	**Course Credits:**	**3**
Course Category:	**E1**	**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.Classify Markov chains in dicrete and continuous time with respect to state diagrams, recurrence			
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, a and whenever applicable derive the distributions as solutions to balance equations			
4. Thoroughly explain the meaning of Markov processes with continous state space, especiall 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

Introductiontostochasticdifferentialequations,Itôcalculus,Fokker-Planckequation,Ornstein-Uhlenbeckprocess.

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

Machine Learning Algorithms			
Course Code:	**CM315**	**Course Credits:**	**3**
Course Category:	**E1**	**Course (U / P)**	**U**
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. To understand and learn regression models, interpret estimates and diagnostic statistics.
2. To understand and learn different classification models and its algorithms
3. To understand and learn clustering methods
4. To generate an ability to build neural networks for solving real life problems
5. To acquire knowledge of Convolution Artificial Neural Networks
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Apply, build and fit regression models for real time problems..
2. Apply and build classification models using SVM and random forest classifiers.
3. Apply and build clustering models using clustering methods and its corresponding algorithms
4. Design and development of certain scientific and commercial application using computational neural network models.
5. Apply text classification and topic modelling methods to solve given problem.

Unit 1 INTRODUCTION

Learning, Types of Learning, Well defined learning problems, Designing a Learning System, History of ML, Introduction of Machine Learning Approaches – (Artificial Neural Network, Clustering, Reinforcement Learning, Decision Tree Learning, Bayesian networks, Support Vector Machine, Genetic Algorithm), Issues in Machine Learning and Data Science Vs Machine Learning.

UNIT 2 REGRESSION: LINEAR REGRESSION AND LOGISTIC REGRESSION

BAYESIAN LEARNING - Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm. SUPPORT VECTOR MACHINE: Introduction, Types of support vector kernel – (Linear kernel, polynomial kernel,and Gaussiankernel), Hyperplane – (Decision surface), Properties of SVM, and Issues in SVM.

Unit 3 DECISION TREE LEARNING

Decision tree learning algorithm, Inductive bias, Inductive inference with decision trees, Entropy and information theory, Information gain, ID-3 Algorithm, Issues in Decision tree learning.

INSTANCE-BASED LEARNING – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning.

Unit 4 ARTIFICIAL NEURAL NETWORKS

Perceptron's, Multilayer perceptron, Gradient descent and the Delta rule, Multilayer networks, Derivation of Backpropagation Algorithm, Generalization, Unsupervised Learning – SOM Algorithm and its variant; DEEP LEARNING - Introduction,concept of convolutional neural network , Types of layers – (Convolutional Layers , Activation function , pooling , fully connected) , Concept of Convolution (1D and 2D) layers, Training of network, Case study of CNN for eg on Diabetic Retinopathy, Building a smart speaker, Self-deriving car etc.

Unit 5 REINFORCEMENT LEARNING

Introduction to Reinforcement Learning , Learning Task,Example of Reinforcement Learning in Practice, Learning Models for Reinforcement – (Markov Decision process , Q Learning - Q Learning function, Q Learning Algorithm), Application of Reinforcement Learning,Introduction to Deep Q Learning. GENETIC ALGORITHMS: Introduction, Components, GA cycle of reproduction, Crossover, Mutation, Genetic Programming, Models of Evolution and Learning, Applications.

Text books:

1. Tom M. Mitchell, ―Machine Learning, McGraw-Hill Education (India) Private Limited, 2013.
2. Ethem Alpaydin, ―Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press 2004.
3. Stephen Marsland, ―Machine Learning: An Algorithmic Perspective, CRC Press, 2009.
4. Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag

DECISION THINKING AND ALGORITHM DESIGN			
CourseCode:	**CM-317**	**CourseCredits:**	**3**
CourseCategory:CC	**E1**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**5U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem.ExamHours:**	**1.5**
TotalNo.of Lectures(L+T):30	**45+00**	**EndSem.ExamHours:**	**3**

COURSE OBJECTIVES
1. Understand different parallel and distributed programming paradigms and algorithms, and gain practicein 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 projectRelate 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 understand a wide variety of learning algorithms and how to evaluate models generated from data.
2. Implementation of Classification and regression algorithms.
3. Implementation and Application of Decision Thinking.
4. To optimize the models learned and report on the expected accuracy that can be achieved by applyingthe models.
5. Apply genetic algorithms to combinatorial optimization problems.

UNIT: -I DECISION MAKING AND COMPUTERIZED SUPPORT

Management 8 Support Systems: An Overview - Decision Making, Systems, Modeling , and Support.

UNIT:-IIDECISION SUPPORT SYSTEMS

Decision Support Systems: An 8 Overview - Modeling and Analysis - Business Intelligence: Data Warehousing,Data Acquisition, Data Mining, Business Analysis, and Visualization - Decision Support System Development.

UNIT: - III COLLABORATIONCOMPUTING

Collaboration, Communication, Enterprise Decision & Support Systems.Collaborative Technologies: GroupSupport Systems - Enterprise Information Systems - knowledge Management.

UNIT: - IV INTELLIGENT DECISION SUPPORT SYSTEMS

Artificial Intelligence 8 and Expert Systems: Knowledge-Based System - Knowledge Acquisition,Representation, and Reasoning - Advanced Intelligent Systems - Intelligent Systems over the Internet.

UNIT:-V IMPLEMENTING IN THE E-BUSINESS ERA

Electronic Commerce - 8 Integration, Impacts, and the Future of the Management Support Systems.

Text Books:

1. Efraim Turban, Jay Aronson E., Ting-Peng Liang, "Decision Support Systems and IntelligentSystems", 7th Edition, Pearson Education, 2006.

STATISTICAL MACHINE LEARNING			
Course Code:	**CM319**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To understand basis of statistics and mathematics for Machine Learning
2. To understand basis of descriptive statistics measures and hypothesis
3. To learn various statistical inference methods
4. To introduce basic concepts and techniques of Machine Learning
5. To learn different linear regression methods used in machine learning.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Apply appropriate statistical measure for machine learning applications
2. Usage of appropriate descriptive statistics measures for statistical analysis
3. Usage of appropriate statistics inference for data analysis
4. Identify types of suitable machine learning techniques
5. Apply regression techniques to machine learning problems

UNIT 1 STATISTICAL INFERENCE I

Types of Statistical Inference, Descriptive Statistics, Inferential Statistics, Importance of Statistical Inference in Machine Learning. Descriptive Statistics, Measures of Central Tendency: Mean, Median, Mode, Mid-range, Measures of Dispersion: Range, Variance, Mean Deviation, Standard Deviation. One sample hypothesis testing, Hypothesis, Testing of Hypothesis, Chi-Square Tests, t-test, ANOVA and ANOCOVA. Pearson Correlation, Bi-variate regression, Multi-variate regression, Chi-square statistics.

UNIT 2 STATISTICAL INFERENCE II

Measure of Relationship: Covariance, Karl Pearson's Coefficient of Correlation, Measures of Position: Percentile, Z-score, Quartiles, Bayes' Theorem, Bayes Classifier, Bayesian network, Discriminative learning with maximum likelihood, Probabilistic models with hidden variables, Linear models, regression analysis, least squares.

UNIT 3 LINEAR ALGEBRA AND CALCULUS

Linear Algebra: Matrix and vector algebra, systems of linear equations using matrices, linear independence, Matrix factorization concept/LU decomposition, Eigen values and eigenvectors. Understanding of calculus: concept of function and derivative, Multivariate calculus: concept, Partial Derivatives, chain rule, the Jacobian and the Hessian.

UNIT 4 INTRODUCTION TO MACHINE LEARNING

What is Machine Learning? Well posed learning problems, Designing a Learning system, Machine Learning types-Supervised learning, unsupervised learning, and Reinforcement Learning, Applications of machine learning, Perspective and Issues in Machine Learning.

UNIT 5 REGRESSION MODEL

Introduction, types of regression. Simple regression- Types, Making predictions, Cost function, Gradient descent, Training, Model evaluation. Multivariable regression : Growing complexity, Normalization,

Making predictions, Initialize weights, Cost function, Gradient descent, Simplifying with matrices, Bias term, Model evaluation

Text Books:

1. Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Education.
2. S.P. Gupta, Statistical Methods, Sultan Chand and Sons, New Delhi, 2009,
3. Kothari C.R., "Research Methodology. New Age International, 2004, 2nd Ed; ISBN:13: 978- 81-224-1522-3.

SEMESTER-6

WEB DEVELOPMENT USING PHP			
Course Code:	**CM302**	**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 Describe fundamentals of web
2. Introduce the creation of static webpage using HTML
3. Describe the function of JavaScript as a dynamic webpage creating tool
4. Outline the principles behind using MySQL as a backend DBMS with PHP
5.Describe the importance of CSS in web development
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Learn and use DHTML and AJAX. Learn the basics of JQuery.
2. Learn about the major vulnerabilities facing web sites and some simple ways to reduce their likelihood
3.Use a MySQL database with PHP to create database applications
4.Design HTML pages and use basic JavaScript code to enhance the pages
5.Develop a complete market-ready database-driven website with PHP and JavaScript and go through the basic phases of the software life cycle

UNITI INTRODUCTION

Internet Standards, Introduction to WWW, WWW Architecture, client and server, web server, web application basic pieces, working of a website, Internet Protocols, Overview of HTTP, HTTP request – response, Generations of dynamic web pages, Front end and backend web development, web content management systems: Wordpress, Joomla, web development life cycle, Guidelines for Indian Government websites.

UNIT II BASICS OF HTML,CSS, JAVASCIPT

HTML and HTML5: Introduction, TML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms. Cascading Style Sheet (CSS): Introduction, Basics of CSS, style types. JavaScript: Introduction, variables, operators, conditionals, looping and validation. Introduction to Jquery, Ajax and XML.

UNIT III INTRODUCTION TO PHP

PHP structure: basic syntax, variables, operators, multiline commands. Expression and control flow in PHP, PHP dynamic linking. PHP functions and Objects, PHP arrays, Practical PHP: Date and time functions, file handling, system calls. Accessing and manipulating database using PHP, Error handling in PHP, generating images with PHP. Cookies, sessions andauthentication.

UNITIV INTRODUCTION TOFRAMEWORK

Introduction of MVC pattern models, MVC works, Configuration Codelgniter, setting up Codelgniter with apache, Environment eg. Enable mod_rewrite, Fetching data, saving and updating data, Deleting data, user defined function in model, Data Validation, controller function, interacting with views, controller variables and parameters, Redirection, Getting post data, working with configuration layout, creating custom layout, Element

and helpers, storing data in cake session, Reading a session data, Delete data from session

UNIT V MYSQL

Databases, Tables, Columns, MySQL Data Type, SELECT, UPDATE and DELETE Statements,PHP and MySQL: Connecting from PHP to MySQL Database, Executing SQL Queries from PHP.

Text Books:

1. LearningPHP,MySQL&JavaScriptwithJQUERY,CSS&HTML5:RobinNixon(O'Reilly)
2. Learning Web Design: A Beginner's Guide to (X)HTML, Style Sheets and Web Graphics: JenniferNiederstRobbins(O'Reilly).

DEEP LEARNING			
Course Code:	**CM304**	**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.Understand the concepts of TensorFlow, its main functions, operations and the execution pipeline
2. Differentiate between machine learning, deep learning and artificial intelligence
3.Learn topics such as convolutional neural networks, recurrent neural networks, training deep networks and high-level interfaces
4. Develop an understanding of the intuition and key mathematical principles behind how neural networks work.
5.Understand the language and fundamental concepts of artificial neural networks Troubleshoot and improve deep learningmodels
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Implement deep learning algorithms, understand neural networks and traverse the layers of data abstraction which will empower the student to understand data more precisely.
2.Build deep learning models in TensorFlow and interpret the results
3.Build own deep learning project
4.Be able to structure and prepare scientific and technical documentation describing project activities.
5.Be able to identify reasonable work goals and estimate the resources required to achieve the objectives.

UNIT 1: Introduction

The Reinforcement Learning problem: evaluative feedback, nonassociative learning, Rewards and returns, Markov Decision Processes, Value functions, optimality and approximation, Bandit Problems: Explore-exploit dilemma, Binary Bandits, Learning automata, exploration schemes Dynamic programming: value iteration, policy iteration, asynchronous DP, generalized policy iteration

UNIT 2:

Monte-Carlo methods: policy evaluation, roll outs, on policy and off policy learning, importance sampling Temporal Difference learning: TD prediction, Optimality of TD(0), SARSA, Q-learning, R-learning, Games and afterstates

UNIT 3:

Eligibility traces: n-step TD prediction, TD (lambda), forward and backward views, Q(lambda), SARSA(lambda), replacing traces and accumulating traces.

UNIT4:

Function Approximation: Value prediction, gradient descent methods, linear function approximation, Control algorithms, Fitted Iterative Methods Policy Gradient methods: non-associative learning - REINFORCE algorithm, exact gradient methods, estimating gradients, approximate policy gradient algorithms, actor-critic methods **UNIT5:**

Hierarchical RL: MAXQ framework, Options framework, HAM framework, Option discovery

algorithms Case studies: Elevator dispatching, Samuel's checker player, TDgammon, Acrobot, Helicopter piloting, Computational Neuroscience.

References

1. R. S. Sutton and A. G. Barto. Reinforcement Learning - An Introduction. MIT Press.1998.
2. Csaba Szepesvari. Algorithms for Reinforcement learning. Morgan & ClaypoolPublishers.
3. Marco Wiering and Martijn van Otterlo, Eds. Reinforcement Learning: State-of-the-Art.Sprinkler.
4. Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach.Pearson.
5. Ian Goodfellow, YoshuaBengio,and Aaron Courville. Deep Learning. MITPress

REINFORCEMENT LEARNING			
Course Code:	**CM306**	**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.Learn how to define RL tasks and the core principals behind the RL, including policies, value functions deriving Bellman equations
2.Understand and work with tabular methods to solve classical control
3.Understand and work with approximate solutions deep Q network based algorithms
4.Explore imitation learning tasks and solutions
5.learn the policy gradient methods from vanilla to more complex
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Implement in code common algorithms following code standards and libraries used in RL
2. Recognize current advanced techniques and applications in RL
3.Identification of suitable learning tasks to which these learning techniques can be applied.
4.Appreciation of some of the current limitations of reinforcement learning techniques.
5.Formulation of decision problems, set up and run computational experiments, evaluation of results from experiments.

UNIT1

Course logistics and overview.Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of machinelearning.

UNIT 2 Probability Primer

Probability concepts - Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

UNIT 3 Markov Decision Process

Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Introduction to and proof of Bellman equations for MRPs, Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, Bellman optimality equations.

UNIT 4 Prediction and Control by Dynamic Programing

Overview of dynamic programming for MDP, definition and formulation of planning in MDPs, principle of optimality, iterative policy evaluation, policy iteration, value iteration, Banach fixed point theorem.

UNIT 5 Policy Gradients

Policy gradient methods, Log-derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.

Text Books

1. Reinforcement Learning – An Introduction (Adaptive Computation and Machine Learning series) by Richard S. Sutton MIT Press; second edition (23 November2018)

2. Deep Reinforcement Learning Hands-On: Apply modern RL methods to practical problems of chatbots, robotics, discrete optimization, web automation, and more, 2nd Edition: Maxim Lapan, Packt Publishing Limited; 2nd edition (31 January2020)

Reference Books

1. Mastering Reinforcement Learning with Python: Build next-generation, self-learning models using reinforcement learning techniques and best practices by Enes Bilgin, Packt Publishing Limited (18 December 2020)
2.
3. Foundations of Deep Reinforcement Learning: Theory and Practice in Python (Addison-Wesley Data & Analytics Series): Laura Graesser Addison-Wesley; 1st edition (4 February2020)

4. Deep Reinforcement Learning with Python: Master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow, 2nd Edition: SudharsanRavichandiranPackt Publishing Limited; 2nd edition (30 September2020)

HUMAN MACHINE INTERACTION			
Course Code:	**CM308**	**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. To learn the foundations of Human Computer Interaction.
2. To become familiar with the design technologies for individuals and persons with disabilities
3. To be aware of mobile HCI.
4. To learn the guidelines for user interface.
5.To analyse how to design user interface
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Design effective dialog for HCI.
2. Design effective HCI for individuals and persons with disabilities
3. Assess the importance of user feedback
4. Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Web sites
5. Develop meaningful user interface.

UNIT I INTROUCTION OF HCI

Human: I/O channels – Memory – Reasoning and problem solving; The Computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms. - Case Studies.

UNIT II DESIGN & SOFTWARE PROCESS

Interactive Design: Basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process: Software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules: principles, standards, guidelines, rules. Evaluation Techniques – Universal Design

UNIT III MODELS AND THEORIES

HCI Models: Cognitive models: Socio-Organizational issues and stakeholder requirements – Communication and collaboration models-Hypertext, Multimedia and WWW.

UNIT IV MOBILE HCI

Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools. - Case Studies.

UNIT V WEB INTERFACE DESIGN

Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow - Case Studies

TEXT BOOKS:

1. AlanDix,JanetFinlay,GregoryAbowd,RussellBeale,—HumanComputerInteraction‖,3rdEdition,Pearson Education, 2004
2. BrianFling,—MobileDesignandDevelopment‖,FirstEdition,O'ReillyMediaInc.,2009
3. BillScottandTheresaNeil,—DesigningWebInterfaces‖,FirstEdition,O'Reilly,2009.

KNOWLEDGE REPRESENTATION			
Course Code:	**CM310**	**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. Differentiate between the concepts data, information and knowledge, and their technologies.
2.Know and know how to use alternative knowledge representation formalisms.
3.Know how to apply knowledge engineering methods for concrete problems
4. Understand which knowledge-based techniques are appropriate for which tasks
5. understand the foundations of KRR and the tradeoff between representation and reasoning
At the end of the course the students should be able to:
1. Represent and solve NP-hard combinatorial problems.
2. Understand the fundamental trade-off between representation power and computational properties of a logic-based representation language
3. Apply KRR systems to their research and challenging problems
4. model complex planning environments using logic-based action description language
5. Solve optimization problems using preferences.

UNIT I Introduction and Concepts
Data, Information and Knowledge; Knowledge Types and Uses; Knowledge Representation, Knowledge Engineering; Syntax andSemantics.

UNIT II KnowledgeRepresentation
Definitions of Knowledge Representation, Architecture, Applications first order logic; Rules and production systems; Object-Oriented Representations; Network Representation; Ontologies.

UNIT III Knowledge Engineering
Knowledge Life-Cycle, Framework of Knowledge Engineering, Applications, Knowledge Audit; Knowledge Acquisition, Detailed Case-Study, Expert System, Architecture of Expert System

UNIT IV Knowledge Representation in the Web
Representing data with HTML; Formalization and representation of information with DTD,XMLSchema, XML; Tools for data and information management on the web with XPath and XSL; Formalization and representation of knowledge with RDF and OWL2.

UNIT IV Logic Programming.
FOL, definite logic programs, Intro to negation & stratified logic programs, Answer Set Semantics,normal logic programs,stable model & answer set semanticsintegrity constraints &

choice, Defeasible reasoning in ASP,Modelling and solving problems in ASP,Practical examples include: Hamiltonian circuits, navigating mazes, fault diagnosis using abduction, generating haiku, understanding conversational implicature using abduction.

Text Books:

1. Ontology Representation, Design Patterns and Ontologies that Make Sense - Volume 197 Frontiers in Artificial Intelligence and Applications ... in Artificial Intelligence and Applications) R.Hoekstra,2009

2. Knowledge representation into Ada parallel processing, National Aeronautics and Space Administration NASA,2018

Reference Books:

1. Foundations of Information and Knowledge Systems11th International Symposium, FoIKS 2020, Dortmund, Germany, February 17–21, 2020, Proceedings Andreas Herzig, JuhaKontinen***2020***

2. Set Theory for Knowledge Representation Set-Theoretical Languages for Reasoning in Knowledge Domains CristianoLongo2012

WEB DEVELOPMENT USING PHP LAB			
Course Code:	**CM382**	**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):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understand best technologies for solving web client/server problems using PHP
2. Analyse& design real time web applications
3. Use PHP for dynamic effects and to validate form input entry
4.Analyze &Develop to Use appropriate client-side or Server-side applications
5. To develop and deploy real time web applications in web servers and in the cloud

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Develop a dynamic webpage by the use of java script and DHTML.
2. Write a well formed / valid XML document using PHP
3. Connect a java program to a DBMS and perform insert, update and delete operations on DBMS table using PHP.
4. Draft a server side application called Servlet to catch form data sent from client, process it and store it on database using PHP
5. Create a server side application to catch form data sent from client and store it on database using PHP

List of Experiments:

1. Basic HTML Tags, TableTags, ListTags, Image Tags, Forms.
2. Implement forms usingHTML,FRAMES,CSS.
3. Install the following on localmachine

- Apache webserver
- Tomcat application serverlocally,
- InstallMySQL
- PHP and configure it to work with Apache web server andMySQL

4. To create an email id for receive and send pictures, documents.
5. To create a simple web file to demonstrate the use of differenttags.
6. To create an html web with different types of frames such as floating frame ,navigation frame & mixed frame.
7. WriteaPHPprogramtostorecurrentdate-timeinaCOOKIEanddisplaythe‘Lastvisitedon’date-timeonthe web page upon reopening of the samepage.

8. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on webpage.
9. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table.Retrieve and display the data based onName.
10. Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with properheadings.

DEEP LEARNING LAB USING PYTHON			
Course Code:	**CM384**	**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):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Implement the various deep learning algorithms in Python.
2.Learn to work with different deep learning frameworks like Keras, Tensor flow, PyTorch, Caffeet
3. Understand complexity of Deep Learning algorithms and their be capable of performing distributed computations.
3. understand modern notions in data analysis oriented computing
4. Be capable of confidently applying common Deep Learning algorithms in practice and implementing their own
5. Be capable of performing experiments in Deep Learning using real-world data
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Troubleshoot and improve deep learning models
2. Understand the concepts of TensorFlow, its main functions, operations and the execution pipeline. & Build own deep learning project
3. Implement deep learning algorithms, understand neural networks and traverse the layers of data abstraction which will empower the student to understand data more precisely
4. Learn topics such as convolutional neural networks, recurrent neural networks, training deep networks and high-level interfaces
5. Build deep learning models in TensorFlow and interpret the results.

List of Practical

1. Basicimageprocessingoperations:Histogramequalization,thresholding,edgedetection,dataaugmentation, morphologicaloperations
2. ImplementSVM/SoftmaxclassifierforCIFAR-10dataset:(i)usingKNN,(ii)using3layerneural network
3. Study the effect of batch normalization and dropout in neural networkclassifier
4. Familiarization of image labelling tools for object detection,segmentation
5. Image segmentation using Mask RCNN, UNet,SegNet
6. Object detection with single-stage and two-stage detectors (Yolo, SSD, FRCNN,etc.)
7. Image Captioning with VanillaRNNs
8. Image Captioning withLSTMs
9. Network Visualization: Saliency maps, ClassVisualization
10. Generative AdversarialNetworks
11. Chatbot using bi-directionalLSTMs
12. Familiarization of cloud based computing like Googlecolab

REINFORCEMENT LEARNING LAB USING Python			
Course Code:	**CM386**	**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):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Design your own algorithm to solve a classical problem from the research community.
2. Learn the difference between the Sarsa, Q-Learning, and Expected Sarsa algorithms.
3. Learn about greedy and epsilon-greedy policies
4. Explore solutions to the Exploration-Exploitation Dilemma.
5. Learn about policies and value functions

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Learn how to build and train neural networks and convolutional neural networks in PyTorch
2. Extend value-based reinforcement learning methods to complex problems using deep neural networks
3. Learn how to implement a Deep Q-Network (DQN), along with Double-DQN, Dueling-DQN, and Prioritized Replay
4. Learn from experts at NVIDIA how to use value-based methods in real-world robotics.
5. Learn how to adapt traditional algorithms to work with continuous spaces.

List of Practicals.

1. Implement Bandit algorithms – UCB,PAC
2. Implement Bandit algorithms –Median Elimination,PolicyGradient
3. Calculate BellmanOptimality
4. Execute Dynamic Programming & TD Methods
5. Write a program to implement Eligibility Traces
6. Implementation of Least Squares Methods
7. Execute all itted Q, DQN & Policy Gradient for FullRL
8. ImplementPOMDPs
9. Execute HierarchicalRL
10. Calcute Function Approximation used in ReinforcementLearning

ELECTIVE II

ARTIFICIAL NEURAL NETWORK			
CourseCode:	**CM 312**	**CourseCredits:**	**3**
CourseCategory:CC	**E2**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem.ExamHours:**	**1**
TotalNo.of Lectures(L+T):30	**45+00**	**EndSem.ExamHours:**	**3**
COURSE OBJECTIVES			
1. To understand the biological neural network and to model equivalent neuron models.			
2. To understand the architecture, learning algorithms.			
3. To know the issues of various feed forward and feedback neural networks.			
4. To explore the Neuro dynamic models for various problems.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand the similarity of Biological networks and Neural networks			
2. Perform the training of neural networks using various learning rules.			
3. Understanding the concepts of forward and backward propagations.			
4. Understand and Construct the Hopfield models.			

UNIT I INTRODUCTION AND ARCHITECTURE

Introduction: A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks Learning Process: Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem, Memory, Adaption, Statistical Nature ofthe Learning Process.

UNIT II LEARNING

Single Layer Perceptrons: Adaptive Filtering Problem, Unconstrained Organization Techniques, Linear Least Square Filters, Least Mean Square Algorithm, Learning Curves, Learning Rate AnnealingTechniques, Perceptron –Convergence Theorem, Relation Between Perceptron and Bayes Classifier for a Gaussian EnvironmentMultilayer Perceptron: Back Propagation Algorithm XOR Problem, Heuristics, Output Representation and Decision Rule, Computer Experiment, Feature Detection.

UNIT III SUPERVISED LEARNING

Back Propagation: Back Propagation and Differentiation, Hessian Matrix, Generalization, Cross Validation, Network Pruning Techniques, Virtues and Limitations of Back Propagation Learning, Accelerated Convergence, Supervised Learning

UNIT IV SELF ORGANIZATION FEATURE MAPS

Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, AdaptivePatter Classification.

UNIT V ATTRACTOR NEURAL NETWORK

Neuro Dynamics: Dynamical Systems, Stability of Equilibrium States, Attractors, Neuro Dynamical Models, Manipulation of Attractors as a Recurrent Network Paradigm Hopfield Models – Hopfield Models, restricted boltzmen machine.

TextBooks:

[1] Neural Networks A Classroom Approach- Satish Kumar, McGraw Hill Education (India) Pvt.

[2] [2]. Neural Networks a Comprehensive Foundations, Simon S Haykin, PHI Ed.,.

[3] Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed. 2006.

Reference Books:

[1] Neural Networks in Computer Inteligance, Li Min Fu TMH 2003 [2]Neural Networks - James A Freeman David M S Kapura Pearson Ed., 2004.

[2] Artificial Neural Networks – B. Vegnanarayana Prentice Hall of India P Ltd 2006.

KNOWLEDGE ENGINEERING			
CourseCode:	**CM 314**	**CourseCredits:**	**3**
CourseCategory:CC	**E2**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**6U**
No.ofLectures+Tutorials(Hrs/Week):	**03+00**	**MidSem.ExamHours:**	**1**
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.

GRAPH THEORY			
Course Code:	**CM316**	**Course Credits:**	**3**
Course Category:	**E2/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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To understand and apply the fundamental concepts in graph theory.
2. To apply graph theory based tools in solving practical problems.
3. To improve the proof writing skills.
4. To introduce the idea of coloring in graphs
5. To have an idea of automorphism groups of graphs

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Apply principles and concepts of graph theory in practical situation.
2. Identify induced subgraphs, cliques , matching, covers in graphs.
3. Determine whether graphs are Hamiltonian and/or Eulerian.
4. Solve problems involving vertex and edge coloring.
5. Solve problems involving vertex and edge connectivity, planarity and crossing numbers.

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

Coloring, 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.

EXPERT SYSTEMS			
Course Code	**CM318**	**Course Credit**	**03**
Course Category	**E2**	**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:**	**01**
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 search strategies.

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 firstorder logic, Inference in first order logic.

UNIT-3UNCERTAIN 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 andactive.

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.

FUZZY LOGIC			
Course Code:	**CM320**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / 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. 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; Arithmetic operations on intervals; Arithmetic Operations on Fuzzy numbers.

UNIT-III - FUZZY RELATIONS AND FUZZY LOGIC

Crisp vs Fuzy relations; Projections and Cylindrical extensions; 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; Non specificity of Crisp Sets; Non specificity 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; Defuzzification to Scalars. Fuzzy inference systems; Mamdani's fuzzy models; Sugeno's fuzzy models; Tsukamoto's fuzzy models; other variants; Applications.

Reference Books

1. Fuzzy Logic Controller for Real Time Networked Control System, B. Sharmila, K. Srinivasan
2. Applying Fuzzy Logic for the Digital Economy and Society, Andreas Meier, Edy Portmann, Luis Terán
3. Robust Control Systems with Genetic Algorithms, Mo Jamshidi, Renato A. Krohling, Leandro dos S. Coelho, Peter J. Fleming

SEMESTER-7

DATA VISUALIZATION			
Course Code:	**CM401**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1.know how to evaluate and criticize data visualizations based on principles of analytic design
2.To understand the position to explore and present their data with visual methods
3.To understand which graphical formats are useful for which types of data and questions
4.To know how to construct compelling visualizations using the free statistics software R
5.Learning appropriate methods for collecting, analyzing, and interpreting numerical information
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Explain principles of visual perception
2.Apply core skills for visual analysis.
3.Apply visualization techniques for various data analysis tasks.
4.Design information dashboard
5.Gaining a basic understanding of the subject (e.g., factual knowledge, methods, principles, generalizations, theories)

UNIT 1: Introduction to data visualization

Acquiring and Visualizing Data, Simultaneous acquisition and visualization, Applications of Data Visualization, Keys factors of Data Visualization (Control of Presentation, Faster and Better JavaScript processing, Rise of HTML5, Lowering the implementation Bar) Exploring the Visual Data Spectrum: charting Primitives (Data Points, Line Charts, Bar Charts, Pie Charts, Area Charts), Exploring advanced Visualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts, Infographics). Making use of HTML5 CANVAS, Integrating SVG.

UNIT 2: Basics of Data Visualization – Tables

Reading Data from Standard text files (.txt, .csv, XML), Displaying JSON content Outputting Basic Table Data (Building a table, Using Semantic Table, Configuring the columns), Assuring Maximum readability (Styling your table, Increasing readability, Adding dynamic Highlighting), Including computations, Using data tables library, relating data table to a chart.

UNIT 3: Visualizing data Programmatically

Creating HTML5 CANVAS Charts (HTML5 Canvas basics, Linear interpolations, A Simple Column Chart, Animations), Starting with Google charts (Google Charts API Basics, A Basic bar chart, A basic Pie chart, Working with Chart Animations)

UNIT 4: Introduction to D3.js

GettingsetupwithD3,Makingselections,changingselection'sattribute,LoadingandfilteringExternaldata: Building a graphic that uses all of the population distribution data, Data formats you can use with D3, Creating a

servertouploadyourdata,D3'sfunctionforloadingdata,DealingwithAsynchronousrequests,Loadinga nd formatting Large Datasets.

UNIT 5: Advanced Data Visualization

Data joins, updates and exits, interactive buttons, Updating charts, Adding transactions, using keys, Introduction, Dashboard design issues and assessment of needs, Considerations for designing dashboard-visual perception, Achieving eloquence, Advantages of Graphics _Library of Graphs, Designing Bullet Graphs, Designing Sparklines, Dashboard Display Media, Critical Design Practices, Putting it all together - Unveiling the dashboard.

Text Books

1. The Visual Display of Quantitative Information (2nd ed.), Edward Tufte (Read pp. 53-77 &107-121)
2. The Elements of Graphing Data (1985) by William S. Cleveland (Read pp. 24-55 &68-88)

References Books:

3. TheTruthfulArt: Data,Charts,and Mapsfor CommunicationbyAlbertoCairo(Readpp.41-65& 121149) 4.Data visualization - Past, present, and future.2-11.
5. Swayne, D., &Klinke, S. (1999). Introduction to the special issue on interactive graphical data analysis: What is interaction? Computational Statistics, 14, 1-6. (Read 3. Interaction: What isit?)
6. M. C., & Roth, S. F. (1996, October). On the semantics of interactive visualizations. In Proceedings IEEE Symposium on Information Visualization'96 (pp. 29-36).IEEE.

APPLIED MACHINE LEARNING			
Course Code:	**CM403**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U**
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.Understand how to correctly prepare input data for use, e.g. feature normalization.
2.Understand how to evaluate and interpret results from scikit-learn estimators.
3.Understand over- and under-fitting and how to detect and prevent these.
4.What data leakage is and how to detect it.
5.Use model selection methods such as cross-validation to tune the choice of model and key parameters.
COURSE OUTCOMES
At the end of the course the students should be able to:
1.Describe the most common types of machine learning problems,
2.Account for why it is important to have informative data and features for the success of machine learning systems
3.explain on a high level how different machine learning models generalize from training examples.
4.Apply a machine learning toolkit in an application relevant to the data science area
5.write the code to implement some machine learning algorithms

UNIT I: Introduction

Outline: Types of ML, ML Process Data exploration (review), ML Process Example K-NN and Accuracy, Feature Normalization, Supervised learning concepts. Regression versus Classification k-NN Regression Linear regression, polynomial feature expansion, measuring error: RSS error, k-fold cross validation, Sci-kit learn datasets Overfitting andunderfitting

UNIT II: Supervised learning and Regularization

Logistic regression, measuring accuracy: ROC, confusion matrix, dealing with categorical and missing data, Regularization: lasso, ridge. Robust regression, Hyper-parameter search, Support vector machines (linear and kernelized): RBF kernels, Multi-class classification, data imputation, data leakage, Decision trees for classification and regression, entropy Boosting, Random forests, gradient boosted decision trees, XGBboost, AdaBoost, feature importance, SVM paper on detecting fraudulent reviews, Naive Bayes,pipelines.

UNIT III: Unsupervised Learning

Unsupervised learning: density estimation, Unsupervised learning: clustering. Agglomerative/tree-based clustering. K-means and variants, Gradient Descent and EM, dimensionality reduction (PCA, multi-dimensional scaling, t-SNE), Evaluation of unsupervised methods, Midterm Examination (tentative).

UNIT IV: Deep Learning

Deep learning, Neural networks, Convolutional NN, Embeddings, Visualizing ConvNets, Sequence problems: Recurrent NN.

UNIT V: Implication of Privacy:

Generative Adversarial networks (GANs), FAT-ML: bias in training and data collection, implications of privacy, Final project presentations (or catch-up), Incentives and Learning, adversarial ML.

Textbooks:

1. IntroductiontoMachineLearningwithPython.A.MuellerandS.Guido.O'Reilly.
2. Deep Learning with Python, by Francois CholletManning

APPLIED MACHINE LEARNING LAB			
Course Code:	**CM 481**	**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):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Identify overfit regression models
2.Compare different regularized regression algorithms and decision tree ensemble algorithms
3.Perform advanced data cleaning, exploration, and visualization
4.Construct training data sets, testing data sets, and model pipelines
5.Explain the confusion matrix and its relation to the ROC curve

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Design experiments with the most common statistical learning models
2.Implement common models using contemporary tools and frameworks
3.Recognize key terms in the discussion of statistical learning.
4.Perform appropriate statistical tests using ML5.

List of Experiments:

1. Getting Starting Install Anaconda Data Pipelines and LearningFrameworks.
2. Write a program to find RandomVariables.
3. Write a program to find BayesRule.
4. Write a program to create a MLE andMAP.
5. Write a program to get the statistical summary and nature of the data of a given dataframe.
6. Write a program to Generative and Discriminative Classifier: NaiveBayes.
7. Write a program to Generative and Discriminative Classifier: LogisticRegression.
8. Write a program to Evaluating StatisticalModels.
9. Write a program to Bias and Variance: LinearRegression.
10. Write a program to Bias and Variance: NearestNeighbor

ELECTIVE III

COMPUTATIONL NEUROSCIENCE			
Course Code:	**CM 405**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
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. The course introduces students to dynamical systems theory for the analysis of neurons and neural circuits.			
2. Students are taught a range of models for neurons and neural circuits,			
3. computational and dynamic properties of these models			
4. implementing and analyzing the behavior of a model for a neural system.			
5describing the kinds of data analysis to be applied to making sense of them.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Students should develop awareness of the philosophical, moral, and ethical issues raised by neuroscience and be able to evaluate arguments critically			
2 Students should demonstrate a proficiency of the structure and function of the nervous system at various levels of organization.			
3 Students should develop critical thinking skills to formulate scientific questions.			
4 Students should understand how to construct testable hypotheses and design scientific investigations that contribute to neuroscience.			
5 Students should understand how to construct testable hypotheses and design scientific investigations that contribute to neuroscience.			

UNIT I Introduction to Neural Modeling.
Introduction to the NEURON simulation environment, Ion flux in membranes, Nernst Planck Equation, Ion-Channels, Excitable membranes, Spiking, Hodgkin Huxley models, Integrate and Fire Neurons.

UNIT II Neural Encoding and Decoding
Spike train statistics, Receptive fields, Linear and Nonlinear models of Receptive fields, Applications of Information Theory in neural coding and decoding, Planar dynamical systems. Limit cycles, oscillators, stability, phase portraits.

UNIT III Plasticity: Adaptation and Learning
Synapses: structure and function, plasticity, Spike Timing Dependent Plasticity (STDP), Learning rules, Supervised and Unsupervised Learning, Classical conditioning, Reinforcement Learning.

UNIT IV Geometry of bursting
The geometry of bursting. Bursts and information processing, Synchronization. Phase oscillators. Phase resetting curves. Resonance and Arnol'd tongues.

UNIT V Network Model

Network models. Memory and pattern classification. The perceptron. Backpropagation neural networks. Hopfield network. Kohonen's self-organizing map. (Selected readings). Alternative: continue with synchronization, noisy resetting curves, Models of the visual pathway: receptive fields, lateral inhibition, orientation tuning, cortical maps.

Text Books:

1.Dayan, Peter, and L. F. Abbott.Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. Cambridge, MA: MIT Press, 2001.

INTELLIGENT MACHINING			
Course Code:	**CM407**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
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. Understand the fundamentals of Artificial Intelligence
2. Learn basics of Intelligent machining, sensors and machining process
3. Understand the design of Intelligent Systems - RTOS
4. Understand the computational methods, optimization and reasoning about physical system
5. Understand implications of Artificial Intelligence in various real time applications
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Acquire the knowledge on the fundamentals of Artificial intelligence and its problem solving approaches
2. Acquire the knowledge on fundamentals of Intelligent Machining and machining process
3. Acquire knowledge on the design of Intelligent Systems and RTOS
4. Acquire knowledge on computational methods and optimization
5. Apply the knowledge on Real time applications

Unit-1 Introduction to Artificial Intelligence

Introduction to Artificial Intelligence and it's techniques, Problem Solving with Artificial Intelligence AI Models, Data acquisition and learning aspects of AI, Problem Solving - Problem Solving Process, Formulating Problems, Problem types and Characteristics, Problem Space and Search, Intelligent Agent Rationality and Rational agent with performance measures, Flexibility and Intelligent Agents, Task Environment and its Properties, Types of Agents, Other aspects of agents, Constraint satisfaction problem (CSP), Crypto Arithmetic puzzles, CSP as a search problem-constraints and representation.

Unit 2 Backtracking and Role of heuristic

CSPbacktracking and Role of heuristic, CSP - Froward Checking and constraint propagation, CSP-Intelligent backtracking. Introduction Intelligent Machining, Basics Open Architecture Machine Control, Manufacturing Automation Protocol, The Evolution of Intelligent Machining,

MOSAIC – NGC, OSACA – SERCOS, Components of Intelligent Machining, Introduction sensors – Machining Process, Sensing and Monitoring , Signal Processing, Transforming Data into Information – Examples, Machining Process Control Practical Uses of Machine Learning.

Unit 3 Learning Process Control Strategies

Machine Learning Process Control Strategies, Programmable Logic Controllers (PLC), Closed Loop Process Control Systems, Introduction to Adaptive Control, Commercially Available Software Representation of Intelligent systems, Control for the Evolution of VLSI Designs, An Object-Oriented Approach, Tools and Techniques for Conceptual Design , Design Compilers, Labelled Interval Calculus, Knowledge Representations for Design Improvisation,

Unit 4 Introduction to RTOS & Memory Management

A knowledge-based Framework for Design, Introduction to RTOS - Hardware Components, Design Principles of RTOS - Interrupt , Processing - task Management, Task Scheduling - Synchronization tools, Task Communication - Memory Management, File System, Tracing and Debugging, Computational methods and optimization, Neural Network Modelling, Fuzzy set theory, Machining Optimization, Objective Functions and Constraints, Optimization Techniques, Reasoning about physical system.

Unit 5 Temporal Qualitative Analysis

Temporal Qualitative Analysis, Reasoning about Geometry, Study of Heuristic knowledge for automatic configuration Generation and Innovation; Case Study - Autonomous Vehicle (Driver Less Car), Flying Drones, Cogito, Alexa , SIRI, Defect Prediction , Wear and Tear Prediction in Mechanical devices, Smarter Home robots, Application of AI in CAD/CAM, Streamlining Drug Discovery, Betterment (Financial Advisor)

References Books:

1. Farid Meziane, Sunil Vadera, Khiary Kobbacy and Nathan Proudlove, "Intelligent Systems in Manufacturing:Current Developments and Future Prospects",
2. How Netflix Uses Analytics To Select Movies, Create Content, and Make Multimillion Dollar Decisions Author: Zach Bulygo
3. Digital Signal Processing: A Practical Guide for Engineers and Scientists, Steven Smith
4. Artifical Intelligent in Engineering Design: Volume 1 , Gerard Meurant, Springer
5. K.C.Wang, " Embedded and Real-Time Operating Systems
6. Sam Siewert, John Pratt," Real-Time Embedded Components and Systems with Linux and RTOS", David Pallai Publisher, 2016.
7. Machining: Fundamentals and Recent Advances, J. Paulo Davim, Springer.
8. Artifical Intelligent in Engineering Design: Volume 2 , Gerard Meurant, Springer

INTRODUCTION TO BRAIN AND NEUROSCIENCE			
Course Code:	**CM 409**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
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 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

Digital Fabrication			
Course Code:	**CM411**	**Course Credits:**	**3**
Course Category: CC	**E3**	**Course (U / P)**	**U**
Course Year (U / P):U	**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):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 51

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)

INTERNET OF THINGS		
Course Code:	**CM413**	**Course Credits:**
Course Category:	**E3**	**Course (U / P)**
Course Year (U / P):	**4U**	**Course Semester (U / P):**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**

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, LegacyDevice 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 DATAANALYTICS 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 V IMPLEMENTING 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

ELECTIVE IV

DIGITAL IMAGE PROCESSING			
Course Code:	**CM415**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for image processing			
2. To study the image enhancement techniques			
3. To study image restoration procedures			
4. To study the image compression procedures.			
5. To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of a digital image processing system.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for image enhancement and image restoration			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

UNIT I INTRODUCTION OF DIGITAL IMAGE PROCESSING

Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Sampling and Quantization, Representing Digital Images (Data structure), Some Basic Relationships Between Pixels- Neighbors and Connectivity of pixels in image, Applications of Image Processing: Medical imaging, Robot vision, Character recognition, Remote Sensing.

UNIT II IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, CombiningSpatial Enhancement Methods.

UNIT III IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

Introduction, Fourier Transform, Discrete Fourier Transform (DFT), properties of DFT , Discrete CosineTransform (DCT), Image filtering in frequency domain.

UNIT IV IMAGE SEGMENTATION

Introduction, Detection of isolated points, line detection, Edge detection, Edge linking, Region based segmentation- Region growing, split and merge technique, local processing, regional processing, Houghtransform, Segmentation using Threshold.

UNIT V IMAGE COMPRESSION

Introduction, coding Redundancy , Inter-pixel redundancy, image compression model, Lossy and Lossless compression, Huffman Coding, Arithmetic Coding, LZW coding, Transform Coding, Sub-image size selection,blocking, DCT implementation using FFT, Run length coding.

TextBooks:

1. Rafael C G., Woods R E. and Eddins S L, Digital Image Processing, Prentice Hall, 3rd edition, 2008

2. Milan Sonka,"Image Processing, analysis and Machine Vision", Thomson Press India Ltd, FourthEdition

Fundamentals of Digital Image Processing- Anil K. Jain, 2nd Edition, Prentice Hall of India

DENSEMBLE LEARNING			
Course Code:	**CM417**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**

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. Ensemble learning aims to achieve better performance with the ensemble of models than with any individual model
2. This requires deciding how to create the models used in the ensemble
3. The objective here is to randomly create samples of training datasets with replacement (subsets of the training data).
4. how best to combine the predictions of the ensemble members
5. The objective of this article is to introduce the concept of ensemble learning and understand the algorithms which use this technique
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Ensembles are predictive models that combine predictions from two or more other models
2. Machine learning predictions follow a similar behavior. Models process given inputs and produce an outcome
3The mechanism for improved performance with ensembles is often the reduction in the variance component of prediction errors made by the contributing models.
4. A minimum benefit of using ensembles is to reduce the spread in the average skill of a predictive model
5. Ensemble learning methods are popular and the go-to technique when the best performance on a predictive modeling project is the most important outcome.

Unit 1 : Introduction

Definition, scope, importance and applications, Model selection techniques , divide and conquer, data Fusion , confidence estimation techniques, relation with machine learning, comparison with other leanings.

Unit 2: Ensemble learning algorithms :

Bagging algorithms and its applications, Boostings and its application with real life example, adaboost algorithm , Stacked generalization and mixture ofexperts algorithms.

Unit 3: Rules and application :

algebric combiners , voting based methods , other combination rules ,Applications: incremental learning , error correcting output codes , feature selection .

Unit 4: Models :

ensemble models and bias variance tradeoff, random forest and bagging models, XGboost model, comparison with other models ,basic requirements of ensemble learning and practical approach of ensemble learning.

Unit 5: **Ensemble Classifiers :**

Definition problems, statistical, computational , representational problem , challenges of model developing , meta – classifiers approach to solve a problems.

Books

1. Supervised and Unsupervised Ensemble Methods and their Applications, 2008.
2. Pattern Classification Using Ensemble Methods, 2010.
3. Ensemble Learning, 2019.
4. Ensemble Methods in Data Mining, 2010.
5. Ensemble Methods, 2012.
6. Ensemble Machine Learning, 2012.

Reference Books

- Hands-On Ensemble Learning with R, 2018.
- Hands-On Ensemble Learning with Python, 2019.
- Ensemble Machine Learning Cookbook, 2019.

PREDICTIVE ANALYSIS			
CourseCode:	**CM 419**	**CourseCredits:**	**3**
CourseCategory: CC	**E4**	**Course(U/P)**	**U**
CourseYear(U/P):U	**4U**	**CourseSemester(U/P):**	**7U**
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 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:

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

5. REFERENCES

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

Embedded Systems			
Course Code:	**CM421**	**Course Credits:**	**3**
Course Category:	E4	**Course (U / P)**	**U**
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. To introduce the Building Blocks of Embedded System			
2. To Educate in Various Embedded Development Strategies			
3. To Introduce Bus Communication in processors, Input/output interfacing.			
4. To impart knowledge in various processor scheduling algorithms.			

5. To introduce Basics of Real time operating system and example tutorials to discuss on one real time operating system tool
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Acquire a basic knowledge about fundamentals of microcontrollers
2. Acquire knowledge about devices and buses used in embedded networking.
3. Develop programming skills in embedded systems for various applications.
4. Acquire knowledge about basic concepts of circuit emulators.
5. Acquire knowledge about Life cycle of embedded design and its testing.

UNIT -I: Introduction to Embedded Systems

Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification, Major Application Areas, Purpose of Embedded Systems, Characteristics and Quality Attributes of Embedded Systems.

UNIT -II: Typical Embedded System:

Core of the Embedded System: General Purpose and Domain Specific Processors, ASICs, PLDs, Commercial Off-The-Shelf Components (COTS), Memory: ROM, RAM, Memory according to the type of Interface, Memory Shadowing, Memory selection for Embedded Systems, Sensors and Actuators, Communication Interface: Onboard and External Communication Interfaces.

UNIT -III: Embedded Firmware:

Reset Circuit, Brown-out Protection Circuit, Oscillator Unit, Real Time Clock, Watchdog Timer, Embedded Firmware Design Approaches and Development Languages.

UNIT -IV: RTOS Based Embedded System Design:

Operating System Basics, Types of Operating Systems, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling.

UNIT -V: Task Communication:

Shared Memory, Message Passing, Remote Procedure Call and Sockets, Task Synchronization: Task Communication/Synchronization Issues, Task Synchronization Techniques, Device Drivers, How to Choose an RTOS.

TEXT BOOKS:

1. Introduction to Embedded Systems - Shibu K.V, Mc Graw Hill.

REFERENCE BOOKS:

1. Embedded Systems - Raj Kamal, TMH.
2. Embedded System Design - Frank Vahid, Tony Givargis, John Wiley.
3. Embedded Systems – Lyla, Pearson, 2013 4. An Embedded Software Primer - David E. Simon, Pearson Education.

Machine Intelligence for Medical Images			
CourseCode:	**CM-423**	**CourseCredits:**	**3**
CourseCategory:CC	**E4**	**Course(U/P)**	**U**
CourseYear(U/P):U	**3U**	**CourseSemester(U/P):**	**7U**
No.ofLectures+Tutorials(Hrs/Week):	**03 + 00**	**MidSem.ExamHours:**	**1.5**
TotalNo.ofLectures(L+ T):45	**45 + 00**	**EndSem.ExamHours:**	**3**

COURSEOBJECTIVES
1. Understand the differences between supervised, unsupervised, weakly, and self-supervised learning.
2. Understand convolutional neural networks (CNN) and can implement CNN in TensorFlow.
3. Use CNN, transformer networks, and transfer learning for image classification.
4 Implement 2D and 3D U-Nets for single-class and multi-class medical image segmentation.
5 Apply CNNGeomtric Network for 2D affine and deformable image registration
COURSEOUTCOMES
At the end of the course the students should be able to:
1. To determine which algorithm is suitable to solve a specific challenge in medical image processing
2. To develop algorithms to solve specific challenges in medical image processing.
3. To apply various segmentation techniques and algorithms in Medical Images
4. Understand the origin of bio-potentials and their physical significance.
5. Compare different techniques of measuring blood pressure, blood flow and volume.

UNIT-I Medical Imaging Basics:
Differences between medical images and natural images, Images as functions, Different imaging modalities, e.g., MRI, CT, ultrasound, PET/SPECT, histopathology, Concept of physical coordinate system, Image visualization in the 3D Slicer software, Introduction to the HiPerGator computing system, Read and write medical images using SimpleITK, Get and set physical information of images, including image dimension, image size, image origin, image spacing, and the direction matrix, Create basic image transformations, including translation, rotation, scaling, and flipping.

UNIT-II Review of Deep Neural Networks:
Artificial neurons, Activation functions, e.g., ReLU, Sigmoid, tanh, Leaky ReLU, Fully connected layers, Differences between supervised, unsupervised, weakly, and self-supervised learning.

UNIT-III Convolutional Neural Networks (CNN) & Image Classicization using TensorFlow
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, Medical applications of image classification, Cross entropy loss, VGG-16, ResNet-101, Image classification with transfer learning.

UNIT-IV Deep Learning Based Image Segmentation & Attention Mechanism:
Transposed convolution, Categorical cross entropy loss vs Dice loss, Fully convolutional networks, U-Net, Query, key, value, Attention function, Dot-product attention and Additive attention Transformer Networks, Self-attention, Multi-head attention, Position encoding.

UNIT-V Introduction to Image registration & Image-to-image Translation:

Clinical applications of image registration, Linear transforms: rigid, affine, Non-linear transforms: thin-plate spline, B-spline, diffeomorphic, Pushforward vs pullback, Interpolators: nearest neighbor, linear, bilinear, Similarity metrics: sum of squared differences, cross correlation, mutual information, SSIM, Challenges in image registration, 2D Image Registration, 3D Image Registration, Paired vs Unpaired image-to-image translation, Pix2Pix, Cycle-GAN, Geometry-consistent GAN, Medical applications of image-to-image translation.

Reference Books:-

1. Shen, Dinggang, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." Annual review of biomedical engineering 19 (2017): 221
2. Litjens, Geert, et al. "A survey on deep learning in medical image analysis." Medical image analysis 42 (2017): 60-88.
3. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.
4. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
5. Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
6. Chen, Jieneng, et al. "Transunet: Transformers make strong encoders for medical image segmentation." arXiv preprint arXiv:2102.04306 (2021).
7. Balakrishnan, Guha, et al. "VoxelMorph: a learning framework for deformable medical image registration." IEEE transactions on medical imaging 38.
8. (2019): 1788-1800. 8. Rocco, Ignacio, Relja Arandjelovic, and Josef Sivic. "Convolutional neural network architecture for geometric matching." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.