SCHOOL
OF
INFORMATION AND COMMUNICATION TECHNOLOGY

COURSE STURCTURE AND DETAILED SYLLABUS

4 YEARS B.TECH

in

COMPUTER SCIENCE AND ENGINEERING

GAUTAM BUDDHA UNIVERSITY
GAUTAM BUDH NAGAR, GREATER NOIDA
2021-2025

SEMESTER – I

Sr. No	Course Code	Courses	L-T-P	Credits
1	MA101	Engineering Mathematics-I	3-1-0	4
2	PH102	Engineering Physics	3-1-0	4
3	EE102	Basic Electrical Engineering	3-1-0	4
4	ME101	Engineering Mechanics	3-1-0	4
5	ES101	Environmental Studies	3-1-0	4
6	PH104	Engineering Physics Lab	0-0-2	1
7	EE104	Basic Electrical Engineering Lab	0-0-2	1
8	EN151	Language Lab	0-0-2	1
9	ME102	Workshop Practice	1-0-2	2
10	GP	General Proficiency	Non Credit	
		Total Credits	**25**	
		Total Contact Hours	**16-5-8=29**	

SEMESTER – II

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS101	Fundamentals of Computer Programming	3-1-0	4
2	CS102	Computer Organization and Architecture	3-1-0	4
3	MA102	Engineering Mathematics-II	3-1-0	4
4	EC101	Basic Electronics Engineering	3-1-0	4
5	CS105	Introduction to Artificial Intelligence	2-0-0	2
6	EN101	English Proficiency	2-0-0	2
7	CE103	Engineering Graphics	1-0-2	2
8	CS181	Computer Programming Lab	0-0-2	1
9	CS183	Computer Organization and Architecture Lab	0-0-2	1
10	EC181	Basic Electronics Engineering Lab	0-0-2	1
11	GP	General Proficiency	Non Credit	
		Total Credits	**29**	
		Total Contact Hours	**17-4-8=29**	

SEMESTER – III

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS201	Internet Technology	3-0-0	3
2	CS203	Concepts of Operating Systems	3-0-0	3
3	CS205	Data Structure and Algorithms	3-0-0	3
4	CS207	Problem Solving using C++	3-0-0	3
5	CS209	Logic Design	3-0-0	3
6	MA201	Engineering Mathematics- III	3-1-0	4
7	CS281	Logic Design Lab	0-0-3	2
8	CS283	Data Structure and Algorithms Lab	0-0-3	2
9	CS285	Object- Oriented Programming Lab	0-0-3	2
10	GP	General Proficiency	Non Credit	
		Total Credits	**25**	
		Total Contact Hours	**18-1-9=28**	

SEMESTER – IV

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS202	Software Engineering	3-0-0	3
2	CS204	Database Management System	3-0-0	3
3	CS206	Java Programming	3-0-0	3
4	CS208	Artificial Intelligence	3-0-0	3
5	CS210	Theory of Automata	3-0-0	3
6	CS 212	Discrete Structure	3-1-0	4
7	CS282	Database Management System Lab	0-0-3	2
8	CS284	Java Programming Lab	0-0-3	2
9	CS286	Artificial Intelligence Lab	0-0-3	2
10	GP	General Proficiency	Non Credit	
		Total Credits	**25**	
		Total Contact Hours	**18-1-9=28**	

SEMESTER V

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS301	Computer Networks	3-0-0	3
2	CS303	Compiler Design	3-0-0	3
3	CS305	Wireless Communication	3-0-0	3
4	CS307	Python	3-1-0	4
5	CS309	**Elective I Computer Graphics**	3-0-0	3
6	CS325	**Elective II Graph Theory**	3-0-0	3
7	CS381	Computer Networks Lab	0-0-3	2
8	CS383	Compiler Design Lab	0-0-3	2
9	CS385	Python Programming Lab	0-0-3	2
10	GP	General Proficiency	Non Credit	
		Total Credits		**25**
		Total Contact Hours	**18-1-9=28**	

Elective I

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS309	Computer Graphics	3-0-0	3
2	CS311	Computer Vision	3-0-0	3
3	CS313	Android Operating System	3-0-0	3
4	CS315	Computer Based Numerical and Statistical Techniques	3-0-0	3
5	CS317	Data Mining		

Elective II

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS319	System Analysis & Design	3-0-0	3
2	CS321	Software Project Management	3-0-0	3
3	CS323	Information Retrieval System	3-0-0	3
4	CS325	Graph Theory	3-0-0	3
5	CS327	Knowledge Engineering	3-0-0	3

SEMESTER VI

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS302	Web Development using PHP	3-0-0	3
2	CS304	Software Testing	3-0-0	3
3	CS306	Analysis and Design of Algorithms	3-1-0	4
4	CS308	Cyber Security	3-0-0	3
5	CS312	**Elective III Ad hoc and Sensor Networks**	3-0-0	3
6	CS328	**Elective IV Big Data Analytics**	3-0-0	3
7	CS382	Web Development using PHP Lab	0-0-3	2
8	CS384	Analysis and Design of Algorithms Lab	0-0-3	2
9	CS386	Cyber Security Lab	0-0-3	2
10	GP	General Proficiency	Non Credit	
		Total Credits		**25**
		Total Contact Hours	**18-1-9=28**	

Elective III

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS310	Digital Image Processing	3-0-0	3
2	CS312	Adhoc & Sensor Networks	3-0-0	3
3	CS314	Expert System	3-0-0	3
4	CS316	Fault tolerant System	3-0-0	3
5	CS318	Mobile Computing		

Elective IV

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS320	Computer security	3-0-0	3
2	CS322	Management Information system	3-0-0	3
3	CS324	Evolutionary Computation	3-0-0	3
4	CS326	Fuzzy logic	3-0-0	3
5	CS328	Big Data Analytics	3-0-0	3

SEMESTER VII

Sr. No	Course Code	Courses	L-T-P	Credits
1	MA401	Modeling and Simulation	3-1-0	4
2	CS401	Internet of Things	3-0-0	3
3	CS403	Soft Computing Techniques	3-0-0	3
4	CS405	Machine Learning	2-0-0	2
5	CS413	**Elective V Cloud Computing**	3-0-0	3
6	CS481	Internet of Things Lab	0-0-3	2
7	CS491	Minor Project	0-0-6	3
8	CS493	Industrial Training	0-0-10	5
9	GP	General Proficiency	Non Credit	
		Total Credits		**25**
		Total Contact Hours	**14-1-19=35**	

Elective V

Sr. No	Course Code	Courses	L-T-P	Credits
1	CS407	Pattern Recognition	3-0-0	3
2	CS409	Robotics	3-0-0	3
3	CS411	Optimization Techniques	3-0-0	3
4	CS413	Cloud Computing	3-0-0	3
5	CS415	Information Security	3-0-0	3

SEMESTER VIII

Sr. No	Course Code	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=49**	

GRAND TOTAL CREDITS- 200

SEM-II

FUNDAMENTALS OF COMPUTER PROGRAMMING			
Course Code:	**CS101**	**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**
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.

UNIT I INTRODUCTION TO COMPUTER AND PROGRAMMING CONCEPTS

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

UNIT II INTRODUCTION TO C PROGRAMMING LANGUAGE

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

UNIT III DATA TYPES AND STRUCTURES

bitwise operators, Pre defined and User defined data types, arrays, declaration and 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 .

UNIT IV FUNDAMENTALS OF POINTERS

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

UNIT V FILE HANDLING IN C AND ENUM

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

Text Books:

1. C Programming by Herbert Shield
2. C Programming Language 2nd Edition by Brian, W Kernighan Pearson Education.
3. Programming in ANSI C by E. Balagurusamy, Tata Mgraw Hill
4. C Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer Prentice HallGale
5. Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling Kindersley India.

COMPUTER ORGANIZATION AND ARCHITECTURE			
Course Code:	**CS102**	**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 point number 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:

1. Patterson, Computer Organisation and Design, Elsevier Pub. 2009
2. William Stalling, " Computer Organization", PHI
3. Vravice,Hamacher & Zaky, "Computer Organization", TMH
4. Mano," Computer System Architecture", PHI
5. John P Hays, " Computer Organization", McGraw Hill
6. Tannenbaum," Structured Computer Organization', PHI
7. P Pal chaudhry, ' Computer Organization & Design', PHI

Introduction to Artificial Intelligence			
Course Code:	**CS105**	**Credits:**	**2**
No. of Lectures (Hrs/Week):	**2**	**Mid Sem Exam Hours:**	**1**
Total No. of Lectures:	**30**	**End Sem Exam Hours:**	**3**

UNIT 1 Introduction

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

UNIT 2 Agents in AI

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

UNIT 3 Searching Techniques and AI Problems

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

UNIT 4 Knowledge Representation

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

UNIT 5 AI Techniques and Applications

Introduction of Machine learning, Introduction of Deep Learning, Introduction to Expert system, Introduction to Natural Language Process, AI in future, AI in social media, AI in Entertainment and Education, AI in Drones, AI in Automated customer support, AI in Personalized shopping Experience, AI in Finance, AI in Smart cars, AI in Travel and Navigation, AI in Smart home Devices, AI in Security and Surveillance, AI in Education, AI in Health Care, AI in E-Commerce.

Reference Books:

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

7. Artificial Intelligence: The Basics By Kevin Warwick

COMPUTER PROGRAMMING LAB			
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):	**1(2 hrs)**	**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 the following:

    a) To find the reverse of a given number.

    b) Calculate factorial of a number using recursion.

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 using switch.

    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 While Loop.

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

   a)
   ```
   *
   **
   ***
   ****
   *****
   ```

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

   c)
   ```
     0
   1 2
   ```

5. Write a program to compute and display the product of two matrices.
6. Write a program to illustrate the difference between call by value and call by reference.
7. Write a program to check whether a given string is palindrome or not.
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 using pointers.
10. Write a program to create a file, input data and display its content.

SEM-III

INTERNET TECHNOLOGY			
Course Code:	**CS201**	**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. Understanding of networking fundamentals.
4. Understanding of learning the process of technology planning.
5. Understanding of 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. Create objects and classes

UNIT I OVERVIEW OF INTERNET AND WEB

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, data security.

UNIT II WEB DESIGN

Key issues in web site design, introduction to HTML, SGML- DTD, DTD elements, attributes, outline of an HTML document, head section- prologue, link, base, meta, script, style, body section- headers, paragraphs, text formatting, linking, internal linking, embedding images, lists, tables, frames, other special tags and characters, XHTML, XML, 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, Language elements, objects of JavaScript, other objects like data, math, string, regular expressions, arrays.

UNIT V ACTIVE SERVER PAGES

Creating interactive applications using active server pages : client and server side script in C#, variables and constants, creating modules, creating objects from classes, ASP‟s object model, arrays, collections, control structures, using request and response objects, integration with database.

Text Books:

1. Raj Kamal, Internet and Web Technologies, TMH, 2005.
2. Monica D‟Souza, Web publishing, 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., Internet and World Wide Web: How to Program, 4 edition, PHI.

CONCEPTS OF OPERATING SYSTEM			
Course Code:	**CS203**	**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. Understanding how Operating System is Important for Computer System.
2. To make aware of different types of Operating System and their services.
3. To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system
4. To know virtual memory concepts and secondary memory management
5. Understanding of Security & protection in Operating System

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understands the different services provided by Operating System at different level
2. They 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. They will learn different memory management techniques like paging, segmentation and demand paging etc.
5. Perform implementation of protection mechanisms in operating system

UNIT I INTRODUCTION TO OPERATING SYSTEM

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

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, classical problem of synchronization, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY MANAGEMENT

Memory management, logical versus physical address space, swapping, contiguous allocation, paging, segmentation, demand paging, page replacement, page replacement algorithms, allocation of frames, thrashing, demand segmentation.

UNIT IV STORAGE MANAGEMENT

File concept, directory structure, protection, file-system structure, allocation method, free-space management, directory implementation.

UNIT V I/O SYSTEMS

I/O hardware, Application of I/O interface, Overview of Kernel I/O subsystem, three types of I/O systems, memory based I/O, I/O based I/O, peripheral based I/O**.**

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.
3. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
4. Stallings William, Operating Systems, PHI, New Delhi, 1997.
5. Madnick and Donavon, Operating Systems, McGraw Hill, International edition, 1978.
6. S. Tanenbaum Modern Operating Systems, , Pearson Education, 3rd edition, 2007..

DATA STRUCTURE AND ALGORITHMS			
Course Code:	**CS205**	**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, extendible hashing.

UNIT II STACK, QUEUE AND LINKED LIST

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

UNIT III TREES

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

UNIT IV SORTING AND SEARCHING

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

UNIT V GRAPHS

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

Text Books:

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

PROBLEM SOLVING USING C++			
Course Code:	**CS207**	**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. Knowledge of Object Oriented Programming paradigm and its features.
2. An understanding of class, object, array, operator, loops and function in C++.
3. Understanding of oops features abstraction, encapsulation, inheritance and polymorphism
4. Understanding of constructor, pointer, static & dynamic memory allocation in C++.
5. Understanding of Files, Files I/O operations and exception handling.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Basic knowledge and understanding of the OOPs concepts and features.
2. Ability to apply OOPs concept for solving the different type of problem.
3. Ability to think how oops can be applied for solving the real life problems and in various projects.
4. To produce efficient software solutions with proper exceptional handling.
5. Can design real life software solutions.

Unit-I

Overview of a Programming Language, Procedural and Object-Oriented Programming paradigms, Structure of a C++ program, Compilation Process of C++ Program, Pre-processor Directives.

Unit-II

Concept of class and Object, Access Control, Primitive Data Types, Arrays, Operators, Identifiers, Keywords, and Literals. Type-casting, Input-Output Statements, Decision Making and Looping Constructs, Recursion, Functions, Command Line Arguments, Libraries.

Unit-III

Abstraction, Encapsulation, Inheritance (Single and Multilevel), Polymorphism (Function Overriding & Overloading, Operator Overloading)

Unit-IV

Pointer and Reference Variables, Runtime Polymorphism, Virtual Function, Constructors and Destructors. Static & Dynamic Memory Allocation.

Unit-V

File I/O Basics, File Operations, Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions.

Text Books:

1. Computer Science: A Structured Approach Using C++. Forouzan & Gilbert (2012) Cengage Learning.
2. C++ Programming: Problem Analysis to Program Design. 7th Edition (2015), D. S. Malik, Cengage Learning.
3. C++: The Complete Reference. 4th Edition. (2003), Schildt, H., Tata McGraw-Hill

LOGIC DESIGN			
Course Code:	**CS209**	**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 understand basic number systems, codes and logical gates.
2. To understand the concepts of Boolean algebra.
3. To understand the use of minimization logic to solve the Boolean logic expressions..
4. To understand the design of combinational and sequential circuits.
5. To understand the state reduction methods for Sequential circuits

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Able to understand number systems and codes.
2. Able to solve Boolean expressions using Minimization methods.
3. Able to design the sequential and combinational circuits.
4. Able to apply state reduction methods to solve sequential circuits.
5. Able to understand the role and working of counters

UNIT 1

Definition of combinational logic, Canonical forms, Generation of switching equations from truth tables, Karnaugh maps-3, 4 and 5 variables, Incompletely specified functions (Don't Care terms), Simplifying Max term equations. Quine-McCluskey minimization technique- Quine-McCluskey using don't care terms, Reduced Prime Implicant Tables, Map entered variables.

UNIT 2

General approach, Decoders-BCD decoders, Encoders. Digital multiplexers-Using multiplexers as Boolean function generators. Adders and subtractors- Cascading full adders, Look ahead carry, Binary comparators. Design methods of building blocks of combinational logics.

UNIT 3

Basic Bistable Element, Latches, SR Latch, Application of SR Latch, A Switch Debouncer, The S R Latch, The gated SR Latch, The gated D Latch, The Master-Slave Flip-Flops (Pulse-Triggered Flip-Flops): The Master-Slave SR Flip-Flops, The Master-Slave JK Flip- Flop, Edge Triggered Flip-Flop: The Positive Edge-Triggered D Flip-Flop, Negative-Edge Triggered D Flip-Flop.

UNIT 4

Characteristic Equations, Registers, Counters - Binary Ripple Counters, Synchronous Binary counters, Counters based on Shift Registers, Design of a Synchronous counters, Design of a Synchronous Mod-6 Counter using clocked JK Flip-Flops Design of a Synchronous Mod-6 Counter using clocked D, T, or SR Flip-Flops

UNIT 5

Introduction, Mealy and Moore Models, State Machine Notation, Synchronous Sequential Circuit Analysis and Design. Construction of state Diagrams, Counter Design.

Text Books:

1. Digital Design by M. Morris Mano
2. Introduction to Logic Circuits & Logic Design with Verilog by Brock J LaMeres
3. Computer System and Architecture by M. Morris Mano
4. Fundamentals of Logic Design, C. H. Roth, L. L. Kinney, 7th edition, Cengage Learning.
5. Fundamentals of Digital Logic & Micro Computer Design, 5TH Edition, M. Rafiquzzaman, John Wiley.

LOGIC DESIGN LAB			
Course Code:	**CS281**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Experiments (Hrs/Week):	**2(3 hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Conduct an experiment to learn the logic design and prototyping process
2. Design a digital module with combinational and sequential logic components to be able to address any problem
3. Use state-of-the-art combinational and sequential logic design methodologies, techniques, and paradigms
4. To understand the use of minimization logic to solve the Boolean logic expressions
5. To understand the basics of various types of memories

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Able to verify the output of various logic gates
2. Able to verify the working of half and full adder circuit
3. Able to design the sequential and combinational circuits.
4. Able to understand the role and working of counters
5. Able to understand and verify various flip flop circuits

List of Experiments:

1. To study and verify the truth table of logic gates
2. To simplify the given expression and to realize it using Basic gates and Universal gates
3. To realize i) Half Adder and Full Adder

   ii) Half Subtractor and Full Subtractor by using Basic gates and NAND gates
4. To design and set up the following circuit using IC 7483.

   i) A 4-bit binary parallel adder.

   ii) A 4-bit binary parallel subtractor.

5. To design and realize the following using IC 7483.

    i) BCD to Excess- 3 Code

    ii) Excess-3 to BCD Code.

6. To realize Binary to Gray code converter and vice versa.

7. To design and set up the following circuit

    i) To design and set up a 4:1 Multiplexer (MUX) using only NAND gates.

8. Truth Table verification of RS Flip Flop,T type flip and D type flip- flop

9. Truth Table verification of JK Flip Flop.

10. To design and test 3-bit binary asynchronous counter using flip-flop IC 7476

DATA STRUCTURE AND ALGORITHMS LAB			
Course Code:	**CS283**	**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)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Introduce the concept of data structures through ADT including List, Stack, Queues .
2. To design and implement various data structure algorithms.
3. To introduce various techniques for representation of the data in the real world.
4. To develop application using data structure algorithms
5. Compute the complexity of various algorithms.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Select appropriate data structures as applied to specified problem definition
2. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
3. Students will be able to implement Linear and Non-Linear data structures.
4. Implement appropriate sorting/searching technique for given problem.
5. Design advanced data structure using Non-Linear data structure

List of Experiments:

1. Run time analysis of Fibonacci Series
2. Study and Application of various data Structure
3. Study and Implementation of Array Based Program
   a. Searching (Linear Search, Binary Search)
   b. Sorting (Bubble, Insertion, Selection, Quick, Merge etc)
   c. Merging
4. Implementation of Link List
   a. Creation of Singly link list, Doubly Linked list
   b. Concatenation of Link list
   c. Insertion and Deletion of node in link list
   d. Splitting the link list into two link list

5. Implementation of STACK and QUEUE with the help of

   a. Array

   b. Link List

6. Implementation of Binary Tree
7. Implementation of Binary Search Tree.
8. Write a program to simulate various traversing Technique
9. Representation and Implementation of Graph

   a. Depth First Search

   b. Breadth First Search

   c. Prims Algorithm

   d. Kruskal's Algorithms

10. Implementation of Hash Table

OBJECT ORIENTED PROGRAMMING LAB			
Course Code:	**CS285**	**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)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Should able to implement constructor overloading, constructor with default argument
2. Understand how to use access modifiers in programming problem
3. Knowledge about implementation of types of inheritance
4. Understanding about destructor and files basic operations
5. Understanding about friend function , virtual function and implementation of function & operator overloading

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Implement the oops concepts in various programming exercises
2. Think of the software solution in view of object oriented paradigm
3. Apply the files usage in C++ programming problem
4. Design an efficient solution for an problem using exception handling mechanism
5. Solve the software project problem using OOPs approach

List of Experiments:

1. A program in C++ to demonstrate the Constructor Overloading , assumed desired program
2. Write a program in C++ demonstrating the public, protected and private parameter and friend value.
3. Write a program in C++ to demonstrate constructor with default argument and destructor in inheritance.
4. Write a program in C++ to demonstrate multiple inheritance.
5. Write a program in C++ to append the content of file. (Assume suitable data)
6. Write a program in C++ to create a file (Assume suitable data)
7. Write a program in C++ to demonstrate virtual function.
8. Write a program to implement an Account class with member function to compute interest, Show balance, withdraw, and deposit amount from the Account.

9. Write a program to implement a sphere class with appropriate member and member function to find surface area and the volume. (Surface = 4 πr2 , and Volume= 4/3 π r3)
10. Write a C++ program to implement a class for complex number with add and multiply as member function. Overload ++ operator to increment a complex number.

SEM-IV

SOFTWARE ENGINEERING			
Course Code:	**CS202**	**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. Knowledge of basic SW engineering methods and practices and application.
2. A general understanding of software process models.
3. Understanding of software requirements and the SRS documents.
4. Understanding of software design process.
5. Understanding of software coding, testing and maintenance.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.
3. Ability to design, develop, maintain and evaluate large-scale software systems.
4. To produce efficient, reliable, robust and cost-effective software solutions.
5. Ability to perform independent research and analysis.

UNIT I SOFTWARE ENGINEERING

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

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

Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

UNIT III SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design, bottom-up design, design methods: data-flow model: data flow

diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, object models: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

UNIT IV SOFTWARE TESTING

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 SOFTWARE MAINTENANCE

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

DATABASE MANAGEMENT SYSTEM			
Course Code:	**CS204**	**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. 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. Students will have an understanding of database concepts and thorough knowledge of database software's.
2. Students will be able to model an application's data requirements using ER diagrams
3. Students will be able to write SQL commands to create tables and query data in a relational DBMS
4. Students will be able to execute various advanced SQL queries related to transactions, concurrency
5. Students will be able to explain the principle of transaction management design.

UNIT I DATABASE SYSTEM

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

UNIT II RELATIONAL MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical data base design, destroying /altering tables and views. relational algebra and calculus: relational algebra, selection and projection set operations,

renaming, joins, division, relational calculus, tuple relational calculus, domain relational calculus, expressive power of algebra and calculus.

UNIT III BASIC SQL QUERY

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

UNIT IV SCHEMA REFINEMENT

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

UNIT V OVERVIEW OF TRANSACTION MANAGEMENT

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

Text Books:

1. Elmasri Navrate, Data Base Management System, Pearson Education, 2008.
2. Raghurama Krishnan, Johannes Gehrke, Data Base Management Systems, TMH, 3rd edition, 2008.
3. C. J. Date, Introduction to Database Systems, Pearson Education, 2009.
4. Silberschatz, Korth, Database System Concepts, McGraw hill, 5th edition, 2005.
5. Rob, Coronel & Thomson, Database Systems Design: Implementation and Management, 2009.

JAVA PROGRAMMING			
Course Code:	**CS206**	**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. To teach principles of object-oriented programming paradigm including abstraction, encapsulation, inheritance, and polymorphism.
2. To impart fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
3. To familiarize the concepts of packages and interfaces
4. To facilitate students in handling exceptions.
5. To 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 OOP implementation

UNIT II DATA TYPE, OPERATORS AND CONTROL STATEMENT

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, wrappers classes, program control statements: decision making and branching: if, if….else, else….if, else if ladder, switch, decision making and looping: while, do….while, for.

UNIT III CLASSES, OBJECTS AND METHODS

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 AND PACKAGES

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 APPLET PROGRAMMING

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. JAVA Beginner‟s guide, Herbert Schildt, Tata McGraw Hill, 2007.
3. Java How to Program, Deitel & Deitel, Prentice-Hall, 1999.

ARTIFICIAL INTELLIGENCE			
Course Code:	**CS208**	**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. To provide a strong foundation of fundamental concepts in Artificial Intelligence
2. To enable the student to apply these techniques in applications which involve perception, reasoning and learning
3. To 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 a system.
5. To enable the student to apply these techniques in applications which involve perception, reasoning and learning

UNIT I INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Basic concept of artificial intelligence (AI), history of AI, AI and consciousness, weak and strong AI, physical symbol system hypothesis, comparison of computer and human skills, practical systems based on AI, development of logic, components of AI.

UNIT II PROBLEM SOLVING THROUGH AI

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

UNIT III SEARCH TECHNIQUES

Use of search in AI problem solution, blind search techniques, heuristic search techniques, concept of heuristic knowledge, designing of the heuristic function, types of heuristic search techniques: generate and test, best first search, problem reduction using AND-OR graph, local search technique, branch and bound search, memory bounded search technique, local beam search, properties of

heuristic search techniques, overestimation and underestimation of heuristic function hill climbing search, simulated annealing search, constraint satisfaction means ends analysis.

UNIT IV INTRODUCTION TO LOGIC

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

UNIT V PROLOG AND LISP

Basic concept of programming languages related to artificial intelligence problems, concept of programming in Logic, basic prolog constructs, atoms, defining the rules, writing small programs in prolog, concept of list processing, basic LISP constructs, writing functions in LISP, some simple programs of LISP.

Text Books:

1. Artificial Intelligence, Elanie Reich: Tata mcgraw Hill publishing house, 2008.
2. Artificial Intelligence, Peterson, TataMcGraw Hill, 2008.
3. Artificial Intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
4. Artificial Intelligence, Winston, PHI publication, 2006

THEORY OF AUTOMATA			
Course Code:	**CS210**	**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. Define various categories of automata (deterministic and nondeterministic finite state automata, and variants of Turing machines)
2. Define 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. Recognize 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, Distinguishing one string from other, Myhill-Nerode Theorem.

UNIT II REGULAR EXPRESSION (RE)

Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen's Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages . Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

UNIT III 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.

UNIT IV PUSH DOWN AUTOMATA (PDA)

Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

UNIT V TURING MACHINES (TM)

Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church's Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory.

Text Books:

1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education.
2. KLP Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata, Languages and Computation", PHI Learning Private Limited, Delhi India.
3. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Publishing house.
4. YN Singh "Mathematical Foundation of Computer Science", New Age International.
5. Malviya, AK "Theory of Computation and Application", BPaperback Publications
6. Papadimitrou, C. and Lewis, CL, "Elements of the Theory of Computation", Pearson Publication.
7. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.

DISCRETE STRUCTURE			
Course Code:	**CS212**	**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. Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.
2. Express a logic sentence in terms of predicates, quantifiers, and logical connectives
3. Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.
4. Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.
5. Apply rules of inference, tests for validity, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.

COURSE OUTCOMES

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

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

JNIT I MATHEMATICAL LOGIC

Statements and notations, connectives, well formed formulas, truth tables, tautology, equivalence implication, normal forms, predicates: predicative logic, free & bound variables, rules of inference, consistency, proof of contradiction, automatic theorem proving, Boolean Algebra: Introduction, Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps(K-Maps).

UNIT II SET THEORY

Set Theory: Introduction, Combination of sets, Multi sets, ordered pairs, Set Identities, Properties of binary relations, equivalence, compatibility and partial ordering relations, Hasse diagram. functions: Operations on functions, inverse function Classification of functions, recursive functions, lattice and

its properties, algebraic structures: algebraic systems examples and general properties, semi groups and monads, groups sub groups‟ homomorphism, isomorphism.

UNIT III ELEMENTARY COMBINATORICS

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

UNIT IV RECURRENCE RELATION

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

UNIT V GRAPH THEORY

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

Text Books :

1. Discrete and Combinational Mathematics- An Applied Introduction-5th Edition – Ralph. P.Grimaldi, Pearson Education
2. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar.P, TMH
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.
4. Discrete Mathematical structures Theory and application-Malik & Sen
5. Discrete Mathematics for Computer science, Garry Haggard and others, Thomson.
6. Logic and Discrete Mathematics, Grass Man & Trembley, Person Education.

DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**CS282**	**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)**	**Mid Sem. Exam Hours:**	
Total No. of Labs (L + T):	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To explain basic database concepts, applications, data models, schemas and instances.
2. To demonstrate the use of constraints and relational algebra operations.
3. To emphasize the importance of normalization in databases.
4. To facilitate students in Database design
5. To familiarize issues of concurrency control and transaction management.

COURSE OUTCOMES
At the end of the course the 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 Language Commands

2. Exercise on Data Manipulation Language and Transaction Control Commands

3. Exercise on Types of Data Constraints

4. Exercise on JOINS (Single-Table) Using Normalization

5. Exercise on JOINS (Multiple-Table) Using Normalization

6. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic

7. Exercise on different Functions (Aggregate, Math and String)

8. Exercise on different types of sub queries

9. Procedures

10. View

JAVA PROGAMMING LAB			
Course Code:	**CS284**	**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)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To prepare students to excel in Object Oriented programming and to succeed as a Java Developer through global rigorous education
2. To make the student learn an object-oriented way of solving problems using java.
3. To make the students to write programs using multithreading concepts and handle exceptions.
4. To make the students to write programs that connects to a database and be able to perform various operations.
5. To 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. Demonstrate an ability to 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. Demonstrate an ability to visualize and work on laboratory and multidisciplinary tasks like console and windows applications both for standalone and Applets programs

List of Experiments:

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

   Class, Command Line Argument, how to enter value through keyboard

2. Write a separate Java Code to implement each of the following data types: Variable, Constant, Arrays, Strings, Vectors, Wrappers Classes, Type Casting

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

   Arithmetic operator, Relational operator, Logical operator, Assignment operator, Increment & Decrement operator, Conditional operator, Bitwise operator, ?: operator

4. Write a separate Java Code to implement each of the following control statements: Decision statement, Loops statement and Branch statements

5. Write a separate Java Code to implement each of the following sorting: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort

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

   Class, Object, Constructors, Method, Method Overloading and Method Overriding

7. Write a separate Java Code to implement each of the following:

   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 and ArrayStoreException

10. Write a separate Java Code to implement the following:

a) Interface

b) Packages and how to import them.

ARTIFICIAL INTELLIGENCE LAB			
Course Code:	**CS286**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Labs (Hrs/Week):	**2(3hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understanding the basic idea of Artificial Intelligence using Prolog
2. To know about 4-Queen problem using prolog.
3. Understanding of the implementation of Fibonacci and factorial series using prolog.
4. Understanding the how to insert and remove the item in the list
5. To learn monkey banana problem using rules in prolog.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Get the basic idea of how to program in prolog and its working environment.
2. Implement 4-Queen problem using prolog.
3. Understand the implementation of Fibonacci and factorial series using prolog.
4. Understand how to solve monkey banana problem using rules in prolog.
5. Understand how to write a program using the rules.

LIST OF EXPERIMENTS:

1. Write a prolog program to find the maximum of two numbers.
2. Write a prolog program to calculate the factorial of a given number. Write a prolog to calculate the nth Fibonacci number.
3. Write a Prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n'th element into every list at all levels.
4. Write a Prolog program, remove-nth(Before, After) that asserts the After list is the Before list with the removal of every n'th item from every list at all levels.
5. Write a program to solve the Monkey Banana problem and 4-Queen problem
6. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
7. Write a Prolog program to implement GCD of two numbers.
8. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.

9. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.

10. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.

SEM-V

COMPUTER NETWORKS			
Course Code:	**CS301**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Describe how computer networks are organized with the concept of layered approach.
2. Implement a simple LAN with hubs, bridges and switches.
3. Analyze the contents in a given Data Link layer packet, based on the layer concept.
4. Describe what classless addressing scheme is.
5. Describe how routing protocols work.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.
2. Have a basic knowledge of the use of cryptography and network security.
3. Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols.
4. Analyse, specify and design the topological and routing strategies for an IP based networking infrastructure
5. Have a working knowledge of datagram and internet socket programming

UNIT I INTRODUCTION AND PHYSICAL LAYER

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

UNIT II SWITCHING AND DATA LINK LAYER

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

UNIT III NETWORK LAYER

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

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

UNIT IV TRANSPORT LAYER

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

UNIT V APPLICATION LAYER

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

Text Books:

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

COMPILER DESIGN			
Course Code:	**CS303**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.
2. Understand relations between computer architecture and how its understanding is useful in design of a compiler.
3. How to construct efficient algorithms for compilers.
4. Provide an understanding of the fundamental principles in compiler design.
5. Learn the process of translating a modern high-level-language to executable code required for compiler construction.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.
2. Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.
3. Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.
4. Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.
5. Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.

UNIT I INTRODUCTION TO COMPILER

Introduction to compiler, phases and passes, bootstrapping, finite state machines and regular expressions and their applications to lexical analysis, optimization of DFA-based pattern matchers implementation of lexical analyzers, lexical-analyzer generator, LEX-compiler, formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC, syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

UNIT II PARSING TECHNIQUE

Parsers, shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic construction of efficient parsers: LR parsers, the canonical collection of LR(0) items, constructing SLR parsing tables, constructing canonical LR parsing tables, constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III SYNTAX-DIRECTED TRANSLATION

Syntax-directed translation schemes, implementation of syntax directed translators, intermediate code, postfix notation, parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser, more about translation: array references in arithmetic expressions, procedures call, declarations and case statements.

UNIT IV SYMBOL TABLES

Data structure for symbols tables, representing scope information, run-time administration: implementation of simple stack allocation scheme, storage allocation in block structured language, Error detection & recovery: lexical phase errors, syntactic phase errors, semantic errors.

UNIT V CODE GENERATION

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, Techniques and Tools", Pearson Education
2. V Raghvan, " Principles of Compiler Design", TMH
3. Kenneth Louden," Compiler Construction", Cengage Learning.

WIRELESS COMMUNICATION			
Course Code:	**CS305**	**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**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To enable students to acquire in-depth knowledge in the field of wireless communication technology with an ability to integrate existing and new knowledge with the advancement of the technology.
2. To develop students to critically analyze the problems in the field of wireless communication technology and find optimal solution.
3. To train students to conduct research and experiments by applying appropriate techniques and tools with an understanding of the limitations for sustainable development of society.
4. To prepare students to act as a member and leader of the team to contribute positively to manage projects efficiently in the field of wireless communication technology.
5. To train students to effectively communicate, write reports, create documentation and make presentations by adhering to appropriate standards.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. An ability to independently carry out research /investigation and development work to solve practical problems.
2. An ability to write and present a substantial technical report/document.
3. A degree of mastery in wireless communication technology at a level higher than the requirements in the appropriate bachelor program.
4. An ability to create, select and apply appropriate techniques and tools to undertake activities in the field of wireless communication technology with an understanding of the limitations.
5. Understand the underlying architecture and use of the latest network models

UNIT I INTRODUCTION TO WIRELESS COMMUNICATION SYSTEM

Evolution of mobile communications, Mobile Radio System around the world, Types of Wireless communication System, Comparison of Common wireless system, Trend in Cellular radio and personal communication. Second generation Cellular Networks, Third Generation (3G) Wireless Networks , Wireless Local Loop(WLL),Wireless Local Area network(WLAN), Bluetooth and Personal Area Networks

UNIT II MOBILE RADIO PROPAGATION MODEL, SMALL SCALE FADING AND DIVERSITY

Large scale path loss:-Free Space Propagation loss equation, Pathloss of NLOS and LOS systems, Reflection, Ray ground reflection model, Diffraction, Scattering, Link budget design, Max. Distance Coverage formula, Empirical formula for path loss, Indoor and outdoor propagation

models, Small scale multipath propagation, Impulse model for multipath channel, Delay spread, Feher's delay spread, upper bound Small scale.

UNIT III MULTIPLE ACCESS TECHNIQUES

Introduction, Comparisons of multiple Access Strategies TDMA,CDMA, FDMA, OFDM , CSMA Protocols. Multiple access for radio pocket system: Pure ALOHA, slotted ALOHA, CSMA and their version : Packet and polling reservation Based multiple access schemes

UNIT IV WIRELESS SYSTEMS

GSM system architecture, Radio interface, Protocols, Localization and calling, Handover, Authentication and security in GSM, GSM speech coding, Concept of spread spectrum, Architecture of IS-95 CDMA system, Air interface, CDMA forward channels, CDMA reverse channels, Soft handoff, CDMA features, Power control in CDMA, Performance of CDMA System, RAKE Receiver, CDMA2000 cellular technology, GPRS system architecture.

UNIT V RECENT TRENDS

Introduction to Wi-Fi, WiMAX, ZigBee Networks, Software Defined Radio, UWB Radio, Wireless Adhoc Network and Mobile Portability, Security issues and challenges in a Wireless network.

Text Books:

1. Wireless Communication, Theodore S. Rappaport, Prentice hall
2. Wireless Communications and Networking,Vijay Garg, Elsevier
3. Wireless digital communication, Kamilo Feher, PHI
4. 4 Mobile Communications Engineering, William C. Y. Lee, Mc Graw Hill Publications
5. 5 Mobile and personal Communication system and services by Rajpandya, IEEE press (PHI).
6. 6 Wireless Communications-T.L.Singh-TMH

PYTHON			
Course Code:	**CS307**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 15**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Master the fundamentals of writing Python scripts.
2. Learn core Python scripting elements such as variables and flow control structures.
3. Discover how to work with lists and sequence data.
4. Write Python functions to facilitate code reuse.
5. Use Python to read and write files.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Problem solving and programming capability.
2. Explain basic principles of Python programming language
3. Implement database and GUI applications.
4. Implement object oriented concepts
5. Define and demonstrate the use of built-in data structures “lists” and “dictionary”

UNIT I PYTHON BASICS, CONDITIONAL &LOOPS

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

UNIT II STRING OBJECTS AND LIST OBJECTS

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

UNIT III TUPLES, SET, DICTIONARIES & FUNCTIONS

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

UNIT IV OOPS CONCEPTS & WORKING WITH FILES

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

UNIT V MODULES, EXCEPTION HANDLING & DATABASE PROGRAMMING

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

Text Books:

1. Head First Python 2e: A Brain-Friendly Guide Paperback – Illustrated, 16 by Paul Barry, Oreilly
2. Python: The Complete Reference Paperback – 20 March 2018 by Martin C. Brown (Author), TMH Publication
3. Let Us Python by Yashavant Kanetkar , 1 January 2019, BPB publication
4. Python Programming, A modular approach , First Edition, By Pearson Publication by Taneja Sheetal and Kumar Naveen , 26 September 2017

COMPUTER NETWORKS LAB			
Course Code:	**CS381**	**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. Practical knowledge of working principles of various communication protocols.
2. Analyze structure and formats of TCP/IP layer protocols.
3. Understanding of networking fundamentals.
4. Understanding of learning the process of Internet of Things applications planning.
5. Understanding of configuration of various end devices, server, routers and switches.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand the practical approach to network communication protocols.
2. Understand network layers, structure/format and role of each network layer.
3. Able to design and implement various network application such as data transmission between client and server, file transfer, real-time multimedia transmission.
4. Understand the various Routing Protocols/Algorithms and Internetworking.
5. Learn to configure server.

List of Experiments:

1. Introduction to transmission media(CAT5, OFC, COAXIAL CABLE Wireless)

2. Introduces network interfaces(Wired and Wireless)

3. Configure and installing a Ethernet(10/100)

4. Performance evaluation of Ethernet(10/100)

5. Topology design(Ring, Bus)

6. Generation of data packet and measurement(CBR, VBR, Poison)

7. Implement the following:

   a) Router configuration

   b) Switch configuration

   c) Server configuration

8. Congestion control of network and QoS of network

9. Protocols and the configuration

10. Security (WEP, WPA) and Qualnet.

COMPILER DESIGN LAB			
Course Code:	**CS383**	**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 implement Lexical Analyzer using Lex tool & Syntax Analyzer or parser using YACC
2. To implement NFA and DFA from a given regular expression
3. To implement front end of the compiler by means of generating Intermediate codes
4. To implement code optimization techniques.
5. To enlighten the student with knowledge base in compiler design and its applications

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Design Lexical analyzer for given language using C and LEX tools.
2. Design and convert BNF rules into YACC form to generate various parsers.
3. Generate machine code from the intermediate code forms
4. Implement Symbol table
5. Demonstrate a working understanding of the process of lexical analysis, parsing and other compiler design aspects.

List of Experiments:

1. Practice of Lex/Yacc of Compiler writing.
2. Write a program to check whether a string belongs to the grammar or not.
3. Write a program to generate a parse tree.
4. Write a program to find leading terminals and trailing terminals
5. Write a program to compute FIRST of non-terminals and FOLLOW of non-terminals.
6. Write a program to check whether a grammar is left recursive and remove left recursion.
7. Write a program to remove left factoring.
8. Write a program to check whether a grammar is Operator precedent.
9. To show all the operations of a stack.
10. To show various operations i.e read, write and modify in a text file.

PYTHON PROGRAMMING LAB			
Course Code:	**CS385**	**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 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 two numbers
2. To read two numbers x and n and print x^n (first write with the use of operator and then write with the help of inbuilt function
3. To input the value of x and n and print the sum of the series:
   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 #vipin # rai

SEM-VI

WEB DEVELOPMENT USING PHP			
Course Code:	**CS302**	**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 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Explain web development Strategies and Protocols governing Web.
2. Design web pages using HTML, XML, CSS and JavaScript
3. Recognize and apply the elements of PHP
4. Introduce with basics concept of Framework
5. Knowledge for database operations using MYSQL and PHP.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand principle of Web page design and about types of websites
2. Visualize and Recognize the basic concept of HTML and application in web designing
3. Understand the elements of PHP
4. Knowledge of Framework
5. Create and manage database using MYSQL and PHP

UNIT I 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 and authentication.

UNIT IV INTRODUCTION TO FRAMEWORK

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. Learning PHP, MySQL & JavaScript with JQUERY, CSS & HTML5: Robin Nixon (O'Reilly)

2. Learning Web Design: A Beginner's Guide to (X)HTML, Style Sheets and Web Graphics: Jennifer Niederst Robbins (O'Reilly).

SOFTWARE TESTING			
Course Code:	**CS304**	**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 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions
5. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. List a range of different software testing techniques and strategies and be able to apply specific(automated) unit testing method to the projects.
2. Distinguish characteristics of structural testing methods.
3. Demonstrate the integration testing which aims to uncover interaction and compatibility problems as early as possible
4. Discuss about the functional and system testing methods.
5. Demonstrate various issues for object oriented testing

UNIT I SOFTWARE TESTING

Essentials of software testing, testing methodology, testing and debugging, software and hardware testing, verification and validation, correctness vs. reliability, challenges in software testing, software testing life cycle (STLC), flow graphs and path testing, transaction flow testing, data flow testing, database testing, web-based testing.

UNIT II SOFTWARE TESTING TECHNIQUES

testing levels: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing; thread testing, regression testing, alpha testing, beta testing, static

testing, dynamic testing, performance testing, ad hoc testing, smoke testing, exhaustive testing, structural testing, mutation testing; Testing Maturity Model (TMM), verification process, defect

tracking, severity and priority, defects, fault, failure, bug, bug life cycle, bug report and bug reporting tools.

UNIT III TEST METRICS AND MEASUREMENT

Purpose of test plan, test plan design, test script, test cases, test management, test case specification, executing test cases, test result analysis. Metrics and measurement, project metrics: effort variance , schedule variance, effort distribution across phases, progress metrics: test defect metrics, development defect metrics, productivity metrics, defect density, defect leakage ratio, Residual Defect Density (RDD), test phase effectiveness, test reports.

UNIT IV SOFTWARE VERIFICATION AND VALIDATION

Verification, methods of verification, types of review on the basis of stage, reviews in testing life cycle, coverage in verification, concerns of verification, validation, coverage in validation, management of Verification and Validation (V & V), software development V & V activities.

UNIT V SOFTWARE TESTING TOOLS

Manual vs. automated testing, functionality and regression testing tool: Win Runner ; load and performance testing tool: Load Runner; web based application testing tool: Quick Test Professional

(QTP); Rational Seed Tool for requirement analysis to testing and project management.

Text Books:

1. Effective Methods for Software Testing, William E. Perry, John Wiley and Sons, 2002.
2. Effective Software Testing: 50 Specific Ways to Improve Your Testing, Dustin, Pearson Education, 2002.
3. An Integrated Approach to Software Engineering, Pankej Jalote, Narosa Publishing House, New Delhi 1997.
4. The Art of Software Testing, Glenford J.Myers, John Wiley & Sons, 1979.
5. Software Testing, Aditya P. Mathur, Pearson Education, 2008.

ANALYSIS & DESIGN OF ALGORITHMS			
Course Code:	**CS306**	**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 + 00**	**End Sem. Exam Hours:**	**3**

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

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

UNIT I BASIC CONCEPT OF ALGORITHMS

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

UNIT II MATHMATICAL ASPECTS AND ANALYSIS OF ALGORITHM

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

UNIT III ANALYSIS OF SORTING AND SEARCHING ALGORITHM

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

UNIT IV ALGORITHM TECHNIQUES

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

UNIT V ALGORITHM DESIGN METHODS

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

Text Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithm", Pearson Education Asia, 2003
2. T.H. Cormen, C.E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithm", PHI Pvt. Ltd., 2001
3. Sara Baase and Allen Van Gelder,"Computer Algorithms-Introduction to the Design and Analysis ", Pearson Education Asia, 2003
4. A. V. Aho, J.E. Hopcroft and J.D. Ullman, "the Design and Analysis of Computer Algorithms", Pearson Education Asia, 2003.

CYBER SECURITY			
Course Code:	**CS308**	**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 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. To prepare students with the technical knowledge and skills needed to protect and defend computer systems and networks.
2. To develop graduates that can plan, implement, and monitor cyber security mechanisms to help ensure the protection of information technology assets.
3. Analyze and resolve security issues in networks and computer systems to secure an IT infrastructure
4. Evaluate and communicate the human role in security systems with an emphasis on ethics, social engineering vulnerabilities and training
5. Interpret and forensically investigate security incident

COURSE OUTCOMES

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

1. Follow a structured model in Security Systems Development Life Cycle (SDLC)
2. Plan, implement and audit operating systems' security in a networked, multi-platform and cross platform environment
3. Protect data and respond to threats that occur over the Internet
4. Design and implement risk analysis, security policies, and damage assessment
5. Detect attack methodology and combat hackers from intrusion or other suspicious attempts at connection to gain unauthorized access to a computer and its resources

UNIT I: CYBER SECURITY FUNDAMENTALS

Overview of cyber security and why it is important; Internet governance: challenges and constraints; cyber threats: cyber warfare, cybercrime, cyber terrorism, cyber espionage; Need for a comprehensive cyber security policy; Need for an international convention on cyber space; Identify trends in cyber security events and protection techniques;

UNIT-II: CYBER SECURITY VULNERABILITIES

Categorize assets, risk, threat and vulnerability; Identify different types of vulnerabilities in a system; Vulnerability in software; Complex network architecture of organization; Open access to organizational data; Weak authentication; Determine the phase of a cyber-attack; Overview of cryptography; Deception; Denial of service filters; Ethical hacking;

UNIT-III: CYBERSPACE AND THE LAW

Computer Ethics and security policies; Cyber security regulations; Role of International Laws; Role of stakeholders (state and private sector) in cyber security; Components of cyber security framework; Cyber security standards; Indian cyber space; National cyber security policy;

UNIT-IV: CYBER FORENSICS

Overview; Why cyber forensics is important; Types of computer forensics; Objective of cyber security forensics investigators; How experts works; Stages of forensics investigation; Techniques and tools used by forensics experts; Advantages of cyber forensics; Incident handling;

UNIT-V: SECURING WEB APPLICATIONS, SERVICES AND SERVER

Introduction; Basic security for HTTP applications and services; Basic security for SOAP (Simple Object Access Protocol)services; Identify management and Web Services; Authorization Patterns; Security Considerations and challenges;

Text Books:

1. Jon Friedman. Mark Bouchard, CISSP. Foreword by John P. Watters to. Cyber Threat Intelligence. Definitive GuideTM. 2015.
2. Scott Roberts, Rebekah Brown. Intelligence-Driven Incident Response: Outwitting the Adversary: O'Reilly 2017.
3. Bob Gourley, The Cyber Threat, Createspace Independent Pub 2014
4. Marjie T Britz, Computer Forensics and Cyber Crime: An Introduction, Pearson Education, 2nd Edition, 2008.
5. Cyberspace and the Law: Your Rights and Duties in the On-Line World By Edward Cavazos and Gavino Morin: MIT Press: 1994.

WEB DEVELOPMENT USING PHP LAB			
Course Code:	**CS382**	**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):	**2(3 hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Perform various logical operations in PHP.
2. Create programs to validate forms in PHP
3. Perform database connectivity using PHP
4. Apply the basic concepts, principles and practices of Web-site development using serverside technologies (PHP &MySQL)
5. Create and manage Blogs,Websites using WordPress

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Develop static web pages using HTML
2. Design dynamic web pages using Java-script , XML and PHP
3. Design dynamic web page using server site programming Ex. ASP/JSP/PHP
4. Working for storing the session and cookies information
5. Understanding for creating the XHTML forms

List of Experiments:

1. Basic HTML Tags,Table Tags,List Tags,Image Tags, Forms .

2. Implement forms using HTML,FRAMES,CSS.

3. Install the following on local machine

- Apache web server
- Tomcat application server locally,
- Install MySQL
- PHP and configure it to work with Apache web server and MySQL

4. To create an email id for receive and send pictures, documents .

5. To create a simple web file to demonstrate the use of different tags.

6. To create an html web with different types of frames such as floating frame, navigation frame & mixed frame.

7. Write a PHP program to store current date-time in a COOKIE and display the ‘Last visited on’ date- time on the web page upon reopening of the same page.

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 web page.

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 on Name.

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 proper headings.

ANALYSIS AND DESIGN OF ALGORITHMS LAB			
Course Code:	**CS384**	**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):	**2(3hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

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

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

PRACTICALS

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

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

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

3. Write a program to obtain the topological ordering of vertices in a given digraph.
4. Implement travelling salesman problem and knapsack problem (0/1).
5. Print all the nodes reachable from a given starting node in a digraph using BFS method.
6.Check whether a given graph is connected or not using DFS method.
7.Write a program to implement binary search using divide and conquer technique
8.Write a program to implement insertion sort using decrease and conquer technique
9. Find minimum cost spanning tree of a given undirected path using a Prim's algorithm.
10. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

CYBER SECURITY LAB			
Course Code:	**CS386**	**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):	**2(3 hrs)**	**Mid Sem. Exam Hours:**	
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Explain & Design computer science terminology related to coding, password protection, social engineering, and network security
2. Learn & implement fundamentals of cryptography and its application to network security.
3. Develop &Understand network security threats, security services, and countermeasures
4. Acquire & Design well known network security protocols such as IPSec, SSL, and WEP
5. Acquire & Develop background on hash functions; authentication; firewalls; intrusion detection techniques.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Develop and implement a java interface for encryption and decryption algorithms i.e., AES, MD5 and RSA algorithm
2. Design and develop a security architecture for an organization.
3. Design operational and strategic cyber security strategies and policies
4. Implement cyber security solutions and use of cyber security, information assurance, and cyber/computer forensics software/tools
5. Comprehend and execute risk management processes, risk treatment methods, and key risk and performance indicators

List of Experiments:

1. Study of different wireless network components and features of any one of the Mobile Security Apps.
2. Study of the features of firewall in providing network security and to set Firewall Security in windows
3. Study of steps to ensure Security of any one web browser (Mozilla Firefox/Google Chrome)
4. Study of System threat attacks - Denial of Service and Sniffing and Spoofing attacks
5. Study of Techniques uses for Web Based Password Capturing.
6. Implementation of S-DES algorithm for data encryption
7. Implementation of Asymmetric Encryption Scheme – RSA.
8. Study of IP based Authentication.
9. To implement the simple substitution technique named Caesar cipher using C language.
10. To write a program to implement the hill cipher substitution techniques

SEM-VII

INTERNET OF THINGS			
Course Code:	**CS401**	**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. Students will be explored to the interconnection and integration of the physical world in IoT.
2. Learning of networking concepts in IoT environment.
3. Understanding of various wireless network, topologies, IoT protocols.
4. Understanding of the importance of security issues in IoT.
5. Implementation of IoT in real life with learning of tools like MATLAB.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand about all concepts of Internet of Things.
2. Understand building blocks of Internet of Things and its characteristics.
3. Learn application protocols for IoT.
4. Able to understand the application areas of IoT.
5. Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.

UNIT I INTRODUCTION TO IOT

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

UNIT II IOT NETWORK ARCHITECTURE AND DESIGN

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

UNIT III NETWORK AND APPLICATION PROTOCOLS FOR IOT

Wireless Communication Technologies: ZigBee, ESP8266, Introduction to sensors and modules - concept, layout, working, applications, Introduction of IoT Development Boards-Node MCU, Arduino, IoT Access Technologies 107IEEE 802.15.4, IEEE 802.15.4g and 802.15.4e, IEEE 1901.2a, IEEE 802.11ah, LoRaWAN, Constrained Devices, Constrained-Node Networks,

Optimizing IP for IoT :From 6LoWPAN to 6Lo, Header Compression, Fragmentation, Mesh Addressing, Mesh-Under Versus Mesh-Over Routing, Authentication and Encryption on Constrained Nodes , Application Protocols for IoT: CoAP, Message Queuing Telemetry Transport (MQTT) .

UNIT IV DATA ANALYTICS AND SECURITY OF IOT

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

UNIT 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

SOFT COMPUTING TECHNIQUES			
Course Code:	**CS403**	**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. The 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

UNIT I 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.

UNIT II FUZZY SETS AND FUZZY LOGIC

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.

UNIT III FUZZY RELATIONS AND FUZZY SYSTEMS

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.

UNIT IV NEURAL NETWORK

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

UNIT V LEARNING FUNDAMENTALS

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.

UNIT VI EVOLUTIONARY ALGORITHMS

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 publication 2004.
3. An Introduction to Genetic Algorithms, Milanie Mitchell, MIT Press 1998.
4. A Genetic Algorithm Tutorial, Darrell Whitley.
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, Particia Melin, Oxford University press, 2003
8. Neural Networks Theory and Application, Oscar Castillo, Wiley Eastern publication
9. Genetic Algorithms in Search, Optimization and Machine Learning, David E Goldberg, Eddison-Wesley, 1988.

MACHINE LEARNING			
Course Code:	**CS405**	**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. 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. Understand 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 ordering of 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-BASED LEARNING – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning

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. 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.Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag.

INTERNET OF THINGS LAB			
Course Code:	**CS481**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
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. Understanding of all basics of sensor networks in IoT environment.
2. Learning of network simulators.
3. Understanding of creation of various wireless network and topologies.
4. Understanding of the importance of energy usage in IoT applications.
5. understand the performance sensors in IoT environment.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understand about Internet of Things applications.
2. Understand the network technology 6LoWPAN.
3. Learn network simulator COOJA, operating system Contiki, and VMWare.
4. Learn to simulate various networks using different sensors.
5. Learn about role of energy reduction in IoT applications.

List of Experiments

1. Basics of sensor networks, IoT, 6LoWPAN nodes (used in IoT applications), OS Contiki, Network Simulator COOJA.

2. Download and Installation of Contiki (OS for IoT), Creation of Virtual Machine, Download and Installation of VM Player.

3. Initialization of Network Simulator COOJA, understanding of all windows on simulator, study the Mote Configuration, Program the Motes so that all motes display “Hello World” on the output window, Change the values in files to display any desired output by all the motes.

4. Create a network topology having 5 motes of similar configuration. Program them to broadcast the data. Capture the broadcasted packets and analyze the values of various headers like IPv6, using analyzer. Repeat the program by changing the transmission range of all motes and observe the effect.

5. Create a complete wireless sensor network (WSN) topology having 6 motes. Configure 1 mote as Border Router and rest of the 5 motes as sender Motes. Go to the browser and check for the values of routing table of your WSN.

6. Repeat the above program on different topology (some motes should not be in the direct range of border router) and check its effect on routing table of border router.

7. Create a Client-Server network topology having 8 motes. Configure 2 motes as server and 6 motes as client. Capture the packets and generate its pcap files. Analyze the captured packets using packet analyzer tool Wireshark.

8. Create a wireless sensor network (WSN) topology having 20 motes of Z1 type. Configure 3 motes as Sink motes and 17 motes as sender motes. Capture the packets in using 6LoWPAN analyzer and also in Wireshark. Analyze the various values of captured packets.

9. Create a wireless sensor network (WSN) topology having 15 motes of Z1 type. Configure 2 motes as Sink motes and 13 motes as sender motes. Calculate complete power consumption of all the motes.

10. Repeat above program on any desired topology. Observe the values of power consumption when motes are in Transmit mode, Receive mode, Sleep mode and Processing mode. Also Calculate power consumption of complete network.

ELECTIVES

COMPUTER GRAPHICS			
Course Code:	**CS309**	**Course Credits:**	**3**
Course Category:	**E1/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics. c.
2. Extract scene with different clipping methods and its transformation to graphics display device.
3. Use of geometric transformations on graphics objects and their application in composite form.
4. Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen

COURSE OUTCOMES

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

1. Explain the core concepts of computer graphics, including viewing, projection, perspective, modelling and transformation in two and three dimensions.
2. Apply the concepts of colour models, lighting and shading models, textures, ray tracing, hidden surface elimination, anti-aliasing, and rendering.
3. Interpret the mathematical foundation of the concepts of computer graphics.
4. Describe the fundamentals of animation, parametric curves and surfaces, and spotlighting.
5. Identify a typical graphics pipeline and apply graphics programming techniques to design and create computer graphics

UNIT 1 BASICS OF COMPUTER GRAPHICS

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

UNIT 2 GRAPHIC DEVICES

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

UNIT 3 TWO DIMENSIONAL TRANSFORMATIONS

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

UNIT 4 GRAPHICAL INPUT TECHNIQUES

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

UNIT 5 COMPUTER GRAPHICS HIDDEN SURFACE REMOVAL

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

Text Books:

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

GRAPH THEORY			
Course Code:	**CS325**	**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.

AD HOC AND SENSOR NETWORKS			
Course Code:	**CS312**	**Course Credits:**	**3**
Course Category:	**E3/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To understand the principles of sensor networks and mobile ad hoc networks, and their impact on protocol design
2. To develop MAC and routing protocols for sensor and mobile networks
3. To develop efficient protocols for sensor and mobile networks
4. To understand and develop information dissemination protocols for sensor and mobile networks
5. To study about the issues pertaining to major obstacles in establishment and efficient management of Ad-hoc and sensor networks.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Describe the unique issues in ad-hoc/sensor networks
2. Describe current technology trends for the implementation and deployment of wireless ad-hoc/sensor networks
3. Discuss the challenges in designing MAC, routing and transport protocols for wireless ad-hoc/sensor networks.
4. Discuss the challenges in designing routing and transport protocols for wireless Ad-hoc/sensor networks
5. Comprehend the various sensor network Platforms, tools and applications.

UNIT II INTRODUCTION :

Fundamentals of Wireless Communication Technology – The Electromagnetic Spectrum – Radio propagation Mechanisms – Characteristics of the Wireless Channel - mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) :concepts and architectures. Applications of Ad Hoc and Sensor networks. Design Challenges in Ad hoc and Sensor Networks.

UNIT II MAC PROTOCOLS FOR AD HOC WIRELESS NETWORKS

Issues in designing a MAC Protocol- Classification of MAC Protocols- Contention based protocols- Contention based protocols with Reservation Mechanisms- Contention based protocols with Scheduling Mechanisms – Multi channel MAC-IEEE 802.11

UNIT III ROUTING PROTOCOLS AND TRANSPORT LAYER IN AD HOC WIRELESS NETWORKS

Issues in designing a routing and Transport Layer protocol for Ad hoc networks- proactive routing, reactive routing (on-demand), hybrid routing- Classification of Transport Layer solutions-TCP over Ad hoc wireless Networks.

UNIT IV WIRELESS SENSOR NETWORKS (WSNS) AND MAC PROTOCOLS :

Single node architecture: hardware and software components of a sensor node – WSN Network architecture: typical network architectures-data relaying and aggregation strategies -MAC layer protocols: self-organizing, Hybrid TDMA/FDMA and CSMA based MAC- IEEE 802.15.4.

UNIT V WSN ROUTING, LOCALIZATION &QOS:

Issues in WSN routing – OLSR- Localization – Indoor and Sensor Network Localization-absolute and relative localization, triangulation- QOS in WSN-Energy Efficient Design-Synchronization-Transport Layer issues.

Text Books:

1. C. Siva Ram Murthy, and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols ", Prentice Hall Professional Technical Reference, 2008.**:**
2. Carlos De Morais Cordeiro, Dharma Prakash Agrawal "Ad Hoc & Sensor Networks: Theory and Applications", World Scientific Publishing Company, 2006.
3. Feng Zhao and Leonides Guibas, "Wireless Sensor Networks", Elsevier Publication – 2002.
4. Holger Karl and Andreas Willig "Protocols and Architectures for Wireless Sensor Networks", Wiley, 2005
5. Kazem Sohraby, Daniel Minoli, & Taieb Znati, "Wireless Sensor Networks-Technology, Protocols, and Applications", John Wiley, 2007.

BIG DATA ANALYTICS			
Course Code:	**CS328**	**Course Credits:**	**3**
Course Category:	**E4/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
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. Understand 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 Implications Access 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 Big Insights Big Data Recommendations

UNIT I 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 DATA PROCESSING

Frameworks: Applications on Big Data Using Pig and Hive, Data processing operators in Pig, Hive services, HiveQL, Querying Data in Hive, fundamentals of HBase and ZooKeeper, IBM InfoSphere BigInsights and Streams.

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. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.

2. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reilly Media, 2012.

3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big Data:

4. Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing, 2012.

CLOUD COMPUTING			
Course Code:	**CS413**	**Course Credits:**	**3**
Course Category:	**E5/DSE**	**Course (U / P)**	**U**
Course Year (U / P):	**5U**	**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. Comprehensive and in-depth knowledge of Cloud Computing concepts,
2. Understand the technologies, architecture and applications
3. Cloud Computing fundamental issues, technologies, applications and implementations.
4. Another objective is to expose the students to frontier areas of Cloud Computing and information systems,
5. while providing sufficient foundations to enable further study and research.

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

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

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

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

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

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

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

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

Text Books:

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