Curriculum for Master of Computer Applications (MCA)
Vidyasagar University, Paschim Midnapore-721102
2018

Program Outcome:

On completion of the MCA (Master of Computer Application) students are able to:

- Work as the Hardware Designers/Engineers with the knowledge of networking concepts.
- Work as IT Sales and Marketing person.
- Work as DTP Operator in small-scale industries.
- Serve as the Programmers or the Software Engineers with the sound knowledge of practical and theoretical concepts for developing software.
- Serve as the IT Officers in Banks and cooperative societies.
- Serve as the System Administrators with thorough knowledge of RDBMS.
- Improve communication and business management skills, especially in providing technical support.
- Develop IT-oriented security issues and protocols.
- Serve as Computer Scientist in research and R & D laboratories.
- Serve as faculty in different general and technical academic institution.
- Serve as Research Staff Member
- Serve as System Analyst in different organization.

FIRST YEAR FIRST SEMESTER (Semester-I)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical Examination	Internal	
MCA-101	Introduction to IT	4	0	0	70	30	4
MCA-102	Computational Mathematics	4	0	0	70	30	4
MCA-103	Digital Logic & Design	4	0	0	70	30	4
MCA-104	Programming in C	4	0	0	70	30	4
MCA-105	Software Architecture	4	0	0	70	30	4

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-191	Digital Logic Lab	0	0	6	70	30	3
MCA-192	Programming Lab	0	0	6	70	30	3
MCA-193	Oral and Written Communication Lab	0	0	6	70	30	3
		20	0	18	560	240	29

Total Period/Week = 38 Total Marks = 800

FIRST YEAR SECOND SEMESTER (Semester-II)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical Examination	Internal	
MCA-201	Data Structure	4	0	0	70	30	4
MCA-202	Computer Architecture	4	0	0	70	30	4
MCA-203	Microprocessor	4	0	0	70	30	4
MCA-204	Numerical methods	4	0	0	70	30	4
MCA-205	Software Project Management	4	0	0	70	30	4

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-291	Data Structure Lab	0	0	6	70	30	3
MCA-292	Microprocessor Lab	0	0	6	70	30	3
MCA-293	Numerical Lab	0	0	6	70	30	3
		20	0	18	560	240	21

Total Period/Week = 38 Total Marks = 800

SECOND YEAR FIRST SEMESTER (Semester-III)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical Examination	Internal	
MCA-301	Database Management System (DBMS)	4	0	0	70	30	4
MCA-302	Theory of Computing	4	0	0	70	30	4
MCA-303	Design and Analysis of Algorithm	4	0	0	70	30	4
MCA-304	Object Oriented Programming	4	0	0	70	30	4
MCA-305	Software Engineering	4	0	0	70	30	4

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-391	DBMS Lab	0	0	6	70	30	3
MCA-392	OOPs Lab	0	0	6	70	30	3
MCA-393	Term Paper-I	0	0	3	0	50	2
MCA394	Industrial Visit	0	0	0	0	50	2
		20	0	15	490	310	30

Total Period/Week = 35 Total Marks = 800

SECOND YEAR SECOND SEMESTER (Semester-IV)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical Examination	Internal	
MCA-401	Computer Graphics	4	0	0	70	30	4
MCA-402	Operating System	4	0	0	70	30	4
MCA-403	Computer Network	4	0	0	70	30	4
MCA-404	Compiler Design	4	0	0	70	30	4
MCA-405	Optimization Technique	4	0	0	70	30	4

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-491	Computer Graphics Lab	0	0	6	70	30	3
MCA-492	Operating System Lab	0	0	6	70	30	3
MCA-493	Computer Network Lab	0	0	3	35	15	2
MCA494	Compiler Design Lab	0	0	3	35	15	2
		20	0	18	560	240	21

Total Period/Week = 38 Total Marks = 800

THIRD YEAR FIRST SEMESTER (Semester-V)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Theoretical Examination	Internal	
MCA-501	Advanced Java	4	0	0	70	30	4
MCA-502	Artificial Intelligence	4	0	0	70	30	4
MCA-503	Web Technology	4	0	0	70	30	4
MCA-504	Elective-I	4	0	0	70	30	4
MCA-505	Elective-II	4	0	0	70	30	4

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-591	Java Lab	0	0	6	70	30	3
MCA-592	AI Lab	0	0	3	35	15	2
MCA-593	Web Technology Lab	0	0	6	70	30	3
MCA-594 (A)	Term Paper-II	0	0	0	0	25	2
MCA-594 (B)	Grand Viva	0	0	0	0	25	
		20	0	15	560	240	30

Total Period/Week = 35 Total Marks = 800

THIRD YEAR SECOND SEMESTER (Semester-VI)

Subjects		Periods per Week			Marks Distribution		Credit Points
Subject Code	Subject Name	L	T	P	Practical Examination	Sessional	
MCA-691	Project Work / Dissertation	0	0	9	0	300	6
MCA-692	Seminar Presentation	0	0	3	0	100	2
MCA-693	Group Discussion	0	0	3	0	100	2
		00	00	15	0	500	10

Total Period/Week = 15 Total Marks = 500

List of electives:

Elective 1, Code: MCA-504

(A) Image Processing
(B) Distributed computing
(C) Mobile Computing
(D) Pattern Recognition
(E) Machine Learning
(F) Soft Computing
(G) Embedded System
(H) Cloud Computing

Elective II, Code : MCA- 505

(A) Multimedia
(B) Data Mining
(C) Parallel Computing
(D) Cryptography and Steganography
(E) Bio Informatics
(F) Natural Language Processing
(G) Computational Geometry
(H) Recent Computing Architecture Trends

Detailed Syllabi of Revised Master of Computer Applications (MCA) Course Vidyasagar University, 2018

FIRST YEAR FIRST SEMESTER (Semester-I)

MCA-101 Introduction to IT [40L]

Objective:

The primary objective of IT is to understand and gather preliminary knowledge of computer and computer system structure. Learner should attain of growing interest to study about computer and how it performs together with the peripheral devices.

Subject outcome:

- The students will able to learn basic concept of computer system.
- They will understand different parts of computer; differentiate between software and hardware, working principle of different input and output devices.
- Learner will represent different number system.
- They will know the details of primary and secondary storage of computer system.
- The students will learn the basic concept of operating system.
- They will know the different security measures of computer system.

Topic Covered:

UNIT – I: Definition of Computer, Features, Parts of Computer System: Hardware, Software, Data, Users, The information processing cycle. Computer Generations. Essential computer hardware: Processing Devices, Memory Devices: RAM, ROM, Input and Output Devices, Storage Devices: Magnetic and Optical Storage, Software: System and Application Software. Input Devices: Keyboard and Mouse: The Standard Keyboard Layout, Using Mouse, Other Data Input Devices: Pen, Touch Screens, Bar Code Readers, OCR Output Devices: CRT Monitors, Printers: Dot Matrix, Ink Jet, Laser Printers, Comparing Printers. **[10L]**

UNIT – II: Number Systems, Computer Arithmetic. Data Processing: The CPU, Memory, Factors Affecting Processing Speed, Bus, Cache Memory. Microcomputer Processor, RISC Processors, Ports: Standard Computer Ports, Serial and Parallel Ports, Specialized Expansion Ports: SCSI, USB, FireWire, MIDI, Expansion Slots and Boards, PC Cards, Plug and Play. **[5L]**

UNIT – III: Secondary Storage Devices: Sequential access devices; Magnetic tapes: Types, Basic Principles of operation. Direct access devices, Magnetic disks: Types, Basic Principles of operation, Advantages, Limitations of magnetic disks. Optical disks: Types, Basic Principles of operation, advantages, Limitations of optical disks. **[8L]**

UNIT – IV : Operating System: Purpose of Operating Systems, Types of Operating System, Managing Hardware: Processing Interrupts, Working with Device Drivers, Utility Software, Backup Utilities, Screen Savers. PC Operating Systems: DOS, Windows 2000 Professional, Windows XP, Macintosh Operating System, UNIX and Linux. **[8L]**

UNIT – V: Productivity Software: Commercial Software, Freeware and Public Domain Software, OpenSource Software. Understanding the Need for Security Measures: Overview, Need for Computer Security, Basic security concepts, Threats to users, hacking, hacking methods. Protective Measures: keeping system safe, avoiding Identity Theft; keeping data secure (Limiting Physical access, Firewall), Managing cookies, spyware & other bugs. **[9L]**

Books:

1. Norton Peter, "Introduction to Computers", 4th Ed., TMH, 2001
2. P. K. Sinha & Priti Sinha , "Computer Fundamentals", BPB Publications, 1992.
3. V. Raja Raman, "Introduction to Computers", PHI,
4. Alex Leon & Mathews Leon, "Introduction to Computers", Vikas Publishing House, 1999.
5. Vikas Gupta, "Comdex Computer Kit", Wiley Dreamtech, Delhi, 2004

MCA-102 Computational Mathematics [40L]

Objective:

Computational Mathematics is the open doors in engineering, business, finance, computing, data sciences, health sciences, environmental sciences and public policy. Recent discoveries in the mathematical sciences have played an essential role in internet search algorithms, disease control, communications technology, climate modeling and much more. Mathematics are among the most important disciplines in today's complex world, in part because they serve as the common language of science.

Subject Outcome:

By the end of this paper a student will:

- have the versatility to work effectively in a broad range of analytic, scientific, government, financial, health, technical and other positions.
- have a broad background in Mathematics, an appreciation of how its various sub-disciplines are related, the ability to use techniques from different areas, and an in-depth knowledge about topics chosen from those offered through the department.
- be mathematically, statistically and numerically literate. In particular, graduates will:
- recognize the importance and value of mathematical thinking, training, and approach to problem solving, on a diverse variety of disciplines;
- be familiar with a variety of examples where mathematics helps accurately explain abstract or physical phenomena;

- recognize and appreciate the connections between theory and applications;
- be able to independently read mathematical literature of various types, including survey articles, scholarly books, and online sources; and
- be life-long learners who are able to independently expand their mathematical expertise when needed, or for interest's sake.

Topic Covered:

UNIT-I: Mathematical reasoning; propositions; negation disjunction and conjunction; implication and equivalence; truth tables; predicates; quantifiers; natural deduction; rules of Inference; methods of proofs; use in program proving; resolution principle; **[10L]**

UNIT-II: Set theory; Paradoxes in set theory; inductive definition of sets and proof by induction; Peano postulates; Relations; representation of relations by graphs; properties of relations; equivalence relations and partitions; Partial orderings; Posets; Linear and well-orderedsets; **[10L]**

UNIT-III: Graph Theory; elements of graph theory, Euler graph, Hamiltonian path, trees, tree traversals, spanning trees; **[4L]**

UNIT-IV: Functions; mappings; injection and surjections; composition of functions; inverse functions; special functions; Peano postulates; pigeonhole principle; recursive function theory; **[6L]**
UNIT-V: Definition and elementary properties of groups, semigroups, monoids, rings, fields, vector spaces and lattices; **[4L]**
UNIT-VI: Elementary combinatorics; counting techniques; recurrence relation; generating functions; **[6L]**

Books:

1. C.L.Liu, Elements of Discrete Mathematics, second edition 1985, McGraw-Hill Book Company. Reprinted 2000.
2. K.H.Rosen, Discrete Mathematics and applications, fifth edition 2003, TataMcGraw Hill publishing Company.
3. J.L.Mott, A.Kandel, T.P .Baker, Discrete Mathematics for Computer Scientists and Mathematicians, second edition 1986, Prentice Hall of India.
4. W.K.Grassmann and J.P.Trembnlay, Logic and Discrete Mathematics, A Computer Science

MCA-103 Digital Logic & Design [40L]

Objectives:

- To understand number representation and conversion between different representation in digital electronic circuits.
- To analyze logic processes and implement logical operations using combinational logic circuits.

- To understand concepts of sequential circuits and to analyze sequential systems in terms of state machines.

Learning Outcome:

After successful completion of the course student will be able to

- Understand different data and number systems, binary representation, codes and their conversions, signed binary number representation with 1's and 2's complement methods.
- Understand binary arithmetic boolean algebra, venn diagram, logic gates and circuits, minimization of logic expressions
- Develop a digital logic and apply it to solve real life problems.
- Analyze, design and implement combinational logic circuits.
- Analyze, design and implement sequential logic circuits.

Topic Covered:

Difference between Analog and Digital Systems. **[1L]**
UNIT-I: Number Systems : Binary number systems, Conversion of binary numbers to decimal numbers and vice-versa, Binary addition, Representation of negative numbers, Binary subtraction, Binary multiplication and division, Octal and hexadecimal numbers, conversion from binary to octal or hexadecimal and vice-versa. **[6L]**
UNIT-II: Binary codes : Binary Coded Decimal (8421 BCD, Excess-3 BCD) , Addition of BCD numbers, Gray code, Error detecting code, Seven-segment display code, other Alphanumeric codes (ASCII, EBCDIC, ISCII, UNICODE). **[6L]**
UNIT-III: Boolean algebra and Logic Gates: Truth Table, AND, OR , NOT and Exclusive-OR operations, Venn Diagram, De Morgan's Theorem, Universal logic operations, Writing Boolean functions from truth table, AND, OR, NOT, NAND, NOR, Exclusive-OR, Exclusive-NOR. **[4L]**
UNIT-IV: Logic Families: Bipolar Logic Families (RTL, DTL, HTL, TTL, ECL), MOS families (MOSFET, CMOS, BiCMOS). **[5L]**
UNIT-V: Minimization of Boolean functions: Karnaugh-Veitch Map method, Quine-Mc-Clausky Method. **[4L]**
UNIT-VI: Combinational digital circuits:** Encoder, Decoder, Multiplexer, Demultiplexer, Magnitude comparator, Parity generator, Parity checker, Half-adder, Full-adder, Sequential adder, Parallel adder, Carry-Look-Ahead adder **[6L]**
UNIT-VII: Sequential digital circuits: Flip-flops, Registers, Up-down counters, asynchronous and synchronous counters, design methodology of sequential circuits. **[8L]**

Books:

1. Digital Design: M. Morris Mano and Michael D. Ciletti, Pearson Education
2. Digital Circuits and Design : S. Salivahanan and S. Arivazhagan, Vikas Publication
3. Engineering Digital Design, R.F.Tinder, Academic Press, Harcourt India Pvt. Ltd.
4. Introduction to Logic design, A.B.Marcovitz, Tata –McGraw-Hill Edition.
5. Computer Systems and Data Analysis, Basu, Nasipuri and Kundu, Narosa, New Delhi.
6. B. Vranesic, " Fundamentals of Digital Logic with VHDL Design", Tata-Mc-Graw-Hill Edition.

7. A. P. Malvino “Digital Principles and Applications”, McGraw Hill International Editions(Fourth
Edition).

MCA-104 Programming in C [45L]

Objective:

The major objective of the course is to provide students with understanding of code organization and functional decomposition with using complex data types.

Outcomes:

After course completion, the students will have the following learning outcomes:

- Understand the fundamentals of C programming.
- Use of loops and decision making statements to solve the problem.
- Implement different of operations on arrays.
- Use functions to solve the given problem.
- Understanding the pointer concept, structures and unions.
- Understanding file concept to show input and output of files in C programming.

Topic Covered:

Fundamentals of C Language: Overview of C, Characters used in C, Identifier, Key Words, Variables, Variables Declaration, Basic Data Types, Additional Data Types, Operators and Expression- (Arithmetic, Relational, Logical, Increment and Decrement, Assignment, Conditional, Bit-wise) , Additional Operator –(sizeof, comma, etc), Structure of a c Program. Managing input and output functions and statement, Formatted Input/Output Functions, Escape Sequences, Character Input/ Output Functions. **[10L]**
Control Statement in C: if-else statement, Nested if Statement, switch Statement etc. **[2L]**
Loop Control in C: for loop, Nested for loop, while loop, do-while loop, goto, break, continue, exit(), etc. **[3L]**
Array : One Dimensional Array, Two Dimensional array. **[2L]**
Handling of character Strings : String Handling Functions in c, Reading / Writing Strings, Additional String Handling Functions, Operations with Characters. **[3L]**
Functions : Define and accessing Functions, Passing arguments, Function prototypes, Recursion, Use of Library functions, Storage Class in C-(auto, static, extern, register). **[6L]**
Structures and unions : structure variables and array, structure with in structure, union, Bit Field. **[4L]**
Pointers : Pointers Declaration, Expression using pointers, pointers as function arguments, pointer arithmetic, pointers with arrays, Dynamic memory allocation, dynamic memory allocation for an array, pointers with string, pointers with structure, pointers with functions, **[10L]**
File management in C: Types of file, file processing, Random file accessing, and errors during file processing. Overview of Pre-processor statements, Program through Command Line Arguments. **[5L]**

Books:

1. Programming with C, Gottfried, TMH
2. C The Complete Reference, Schildt, TMH
3. Practical C Programming,3rd Ed,Oualline,SPD/O'REILLY
4. A First Course in programming with C, Jeyapoovan, VIKAS
5. The C answer Book, Tondo,2nd Ed, PHI
6. C Programming Made Easy, Raja Ram, SCITECH
7. Projects Using C, Varalaxmi,SCITECH
8. Mastering Algorithms With C,Loudan, SPD/O'REILLY

MCA-105 Software Architecture [40L]

Objective:

- Knowledge of basic SW engineering methods and practices, and their appropriate application.
- Describe software engineering layered technology and Process frame work.
- A general understanding of software process models such as the waterfall and evolutionary models.
- Understanding of software requirements and the SRS documents.
- Understanding of the role of project management including planning, scheduling, risk management, etc.
- Describe data models, object models, context models and behavioural models.
- Understanding of different software architectural styles.
- Understanding of implementation issues such as modularity and coding standards.
- Understanding of approaches to verification and validation including static analysis, and reviews. 10. Understanding of software testing approaches such as unit testing and integration testing.
- Describe software measurement and software risks.
- Understanding of software evolution and related issues such as version management.
- Understanding on quality control and how to ensure good quality software.

Outcome:

- Basic knowledge and understanding of the analysis and design of complex systems.
- Ability to apply software engineering principles and techniques.
- Ability to develop, maintain and evaluate large-scale software systems.
- To produce efficient, reliable, robust and cost-effective software solutions.
- Ability to perform independent research and analysis.
- To communicate and coordinate competently by listening, speaking, reading and writing english for technical and general purposes.
- Ability to work as an effective member or leader of software engineering teams.
- To manage time, processes and resources effectively by prioritising competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.

- Ability to understand and meet ethical standards and legal responsibilities.

Topic Covered:

Introduction: The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle; What makes a "good" architecture? What software architecture is and what it is not; Other points of view; Architectural patterns, reference models and reference architectures; Importance of software architecture; Architectural structures and views. **[6L]**

Architectural Styles and Case Studies :Architectural styles; Pipes and filters; Data abstraction and object-oriented organization; Event-based,implicit invocation; Layered systems; Repositories; Interpreters; Process control; Other familiar architectures; Heterogeneous architectures. Case Studies: Keyword in Context; Instrumentation software; Mobile robotics; Cruise control; Three vignettes in mixed style. **[8L]**

Quality: Functionality and architecture; Architecture and quality attributes; System quality attributes; Quality attribute scenarios in practice; Other system quality attributes; Business qualities; Architecture qualities. Achieving Quality: Introducing tactics; Availability tactics; Modifiability tactics; Performance tactics; Security tactics; Testability tactics; Usability tactics; Relationship of tactics to architectural patterns; Architectural patterns and styles. **[6L]**

Architectural Patterns – 1: Introduction; From mud to structure: Layers, Pipes and Filters, Blackboard. **Architectural Patterns – 2** : Distributed Systems: Broker; Interactive Systems: MVC, Presentation-Abstraction-Control. **Architectural Patterns – 3** :Adaptable Systems: Microkernel; Reflection. **[10L]**

Designing and Documenting Software Architecture: Architecture in the life cycle; Designing the architecture; Forming the team structure; Creating a skeletal system. Uses of architectural documentation; Views; Choosing the relevant views; Documenting a view; Documentation across views. **[10L]**

Books:

1. Documenting Software Architectures: Views and Beyond, P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, MA: Addison-Wesley, 2003.
2. Software Architecture in Practice, L. Bass, P. Clements & R. Kazman, Addison Wesley.
3. Architecting Software Intensive Systems: A Practitioner's Guide, A. Lattanze, Boca Raton, FL: Auerbach Publishing, 2008.
4. Component-Based Software Engineering, Edited by A. W. Brown, IEEE Computer Society
5. Design Patterns: Elements of Reusable Object-Oriented Software,Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesle
6. Design Patterns for Object-Oriented Software Development, Wolfgang Pree, Addison-Wesley Longman Seamless

MCA-191 Digital Logic Lab

Objectives:

- Realization of gates.
- Design of combinational circuits.
- Design of sequential circuits.

Learning Outcome:

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

- Realize the working principle of gates
- Different Code conversion procedures
- Design of combinational circuits to drive seven-segment display
- Design of combinational circuits like - parity generation and checker, comparator, Adder/Subtractor, multiplexer etc.
- Design of sequential circuits like- flip flops, counter, register etc.

Topic Covered:

Realization of NOT, OR, AND, XOR, XNOR gates using universal gates
Gray to Binary conversion & vice-versa.
Code conversion between BCD and EXCESS-3
ODD and even parity generation and checking.
4-bit comparator circuit
Design of combinational circuit to drive seven-segment display
Design of combinational circuits using multiplexer
Adder/Subtractor circuits using Full-Adder using IC and/ or logic gates., BCD Adder circuit using IC and/ or logic gates
Realization of RS, JK, and D flip flops using Universal logic gates
Realization of Asynchronous up/down counter
Realization of Synchronous Mod-N counter
Digital to Analog conversion

Books:

1. Givone: Digital Principles &design ,TMH
2. Digital Electronics – Dr. Saroj Rangnekar , ISTE/EXCEL BOOKS
3. Malvino:Digital Principles &application, TMH
4. Jain :Modern Digital Electronics 2/e, TMH
5. Marcovitz:Introduction to logic Design- Tata Mcgraw-hill
6. Digital Integrated Electronics- H.Taub & D.Shilling, Mc Graw Hill
7. Digital Technology- Virendra Kumar, New Age
8. Digital Logic Design- Morries Mano, PHI
9. Yarbrough- Digital Logic,Vikas
10. Salivahan- Digital Circuits and Design, Vikas

MCA-192 Programming Lab

Objective:

The major objective of the course is to enhance the C programming skills of the students. The concept of a structured and procedural programming language helps them to implement various complex real world problems.

Outcomes:

After course completion the students will have the following learning outcomes:

- Illustrate the flowchart and design an algorithm for a given problem.
- Understand basic structure of C programming, declaration and uses of variables.
- Write C programs using operators.
- Using conditional, iterative, and recursive statements to write C programs.
- Writing C programs using pointers to access arrays, strings and functions.
- Dynamic memory allocation and management in a C program.
- Exercise user defined data types.
- Using file concept to read and write to the file using C programs.

Topic Covered:

Experiments should include but not limited to:

1. Write a C program to find the factorial of a positive integer.
2. Write a C program to find the roots of a quadratic equation.
3. Write a C program to determine if the given number is a prime number or not.
4. Write a C program to generate the first n terms of the Fibonacci sequence.
5. Write a C program to construct a pyramid of numbers.
6. Write a C program to calculate the following Sum: Sum = 1 - x2/2! + x4/4! - x6/6! + x8/8! - x10/10!
7. The least common multiple (LCM) of two positive integers a and b is the smallest integer that is evenly divisible by both a and b. Write a C program that reads two integers and calls LCM (a, b) function that takes two integer arguments and returns their LCM. The LCM (a, b) function should calculate the least common multiple by calling the GCD (a, b) function and using the following relation: LCM(a, b) = ab / GCD(a, b)
8. Write a C program that reads two integers n and r to compute the ncr value using the following relation: ncr (n, r) = n! / r! (n-r)! . Use a function for computing the factorial value of an integer.
9. Write C program that reads two integers x and n and calls a recursive function to compute x^n

10. Write a C program that uses a recursive function to solve the Towers of Hanoi problem.
11. Write a C program that reads two integers and calls a recursive function to compute ncr value.
12. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user using Sieve of Eratosthenes algorithm.
13. Write a C program that uses non recursive function to search for a Key value in a given list of integers. Use linear search method.
14. Write a menu-driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.
15. Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers. Use binary search method.
16. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.
17. Write a C program that reads two matrices and uses functions to perform the following:
18. Write a C program that uses a non recursive function to determine if the given string is a palindrome or not.
19. Write a C program to replace a substring with another in a given line of text.
20. Write a C program that reads 15 names each of up to 30 characters, stores them in an array, and uses an array of pointers to display them in ascending (ie. alphabetical) order.
21. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.
22. Write a C program to convert a positive integer to a roman numeral. Ex. 11 is converted to XI.
23. Write a C program to display the contents of a file to standard output device.
24. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
25. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command-line arguments.
26. Write a C program to compare two files, printing the first line where they differ.
27. Write a C program to change the nth character (byte) in a text file. Use fseek function.
28. Write a C program to reverse the first n characters in a file. The file name and n are specified on the command line. Use fseek function.
29. Write a C program to merge two files into a third file (i.e., the contents of the firs t file followed by those of the second are put in the third file).
30. Define a macro that finds the maximum of two numbers. Write a C program that uses the macro and prints the maximum of two numbers.

Books:

1. Mastering C, K.R. Venugopal and S.R. Prasad, TMH Publishers.
2. Computer Programming in C, V. Rajaraman, PHI.
3. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
4. C++: The complete reference, H. Schildt, TMH Publishers.

MCA-193 Oral and Written Communication Lab

Outcomes:

Students will be able to:

- Effectively communicate through the written word for a wide variety of purposes and audiences
- Accurately convey the intent of their message when writing
- Structure written communication that is clear, logical, and easy to follow
- Use correct mechanics such as grammar, spelling, and punctuation
- Select and incorporate appropriate supporting materials
- Effectively adjust their writing styles to appropriately address the audience

Topic Covered:

1. Introductory lecture is to be given to the students so that they get a clear idea of the syllabus and understand the need for having such a practice lab in the first place.
2. Conversion practice is done on given situation topics. The students are also made to listen to pre-recorded cassettes produced by British Council and also by the Universities of Oxford and Cambridge.
3. Group Discussions:- The students are made to understand the difference between the language of conversion and group discussion. Strategies of such discussions are to teach to them. It is also helpful to use videocassettes produced by the U.G.C. on topics like group-discussion. After wards the class is divided into groups and the students have to discuss on given topics on current socio-economic-political-educational importance.
4. Interview sessions-students are taught the do's and don'ts of facing a successful interview. They then have to face rigorous practices of mock-interviews. There simulations of real life interview sessions where students have to face an interview panel.
5. Presentations: The secrets of an effective presentation are taught to the students. Then each and every student has to make lab presentations with the help of the Overhead projector/ using power point presentation and other audio-visual aids in the laboratory. They also have to face the question answer sessions at the end of their presentation.
6. Classes are also allotted to prepare the students for competitive examinations like the T.O.E.F.L. by making the students listen to specially produced C.D. cassettes of such examinations.
The overall aim of this course is to inculcate a sense of confidence in the students and help them to become good communicators in their social as well as professional lives.

Books:

1. Sharma—Business Correspondence & Report Writing, TMH
2. Prasad—Group Discussion & Interview (With Audio Cassette) , TMH
3. Sashi Kumar—Spoken English (with Cassette) , TMH

FIRST YEAR SECOND SEMESTER (Semester-II)

MCA-201 Data Structure [40 L]

Objective:

Students will try to learn:

- Understand and remember algorithms and its analysis procedure.
- Introduce the concept of data structures through ADT including List, Stack, Queues.
- To design and implement various data structure algorithms.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structure algorithms.
- Compute the complexity of various algorithms

Outcome:

Students will able to:

- Select appropriate data structures as applied to specified problem definition.
- Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
- Students will be able to implement Linear and Non-Linear data structures.
- Implement appropriate sorting/searching technique for given problem.
- Design advance data structure using NonLinear data structure.
- Determine and analyze the complexity of given Algorithms.

Topic Covered:

Introduction – Data and Information - Program Structures – Abstract Data Type – Data Structure - Static and Dynamic Data Structures **[2L]**

Array as a Data Structure – Representation of Polynomials and Sparse Matrix - Linear List – Implementation using array **[4L]**

Review of Pointers and Functions **[1L]**

Linked List – Single and Double Linked List – Applications **[3L]**

Concepts of Algorithm Design Techniques – Divide and Conquer – Greedy – Dynamic **Programming** – Backtracking – Branch and Bound – Examples **[3L]**

Concepts of Algorithm Analysis – Performance Measurement and Analysis – Time Complexity and Space Complexity – Introduction to Order functions – Examples of Analysis **[3L]**

Searching and Sorting Algorithms – Linear and Binary Search – Sorting – Insertion, Selection, Merge, Quick, Heap, Bucket – Stable sorting **[3L]**

Stack and Queue – Implementations using Arrays and Linked List – Applications – Expression Evaluation and Conversions **[3L]**

Recursion – Types of Recursion – Examples – Implementation using stack **[3L]**

Trees – Binary Trees – Binary Search Tree – Balanced Trees – 2-3 Tree – B-Tree – B+-Tree **[10L]**

Graphs – Adjacency Matrix and List – Graph Search Algorithms – Spanning Tree Algorithms – Shortest Path Algorithms – Transitive Closure Matrix **[3L]**

Hashing – Terminologies – Hash functions – Collision Resolution Strategies – Types of Hashing **[2L]**

Books:

1. Fundamentals of Data Structures in C by Horowitz, Sahni & Anderson-Freed, 2e Universal Press
2. Data Structures and Algorithm Analysis in C by Mark Alan Weiss, 2nd ed., Pearson Education
3. Data Structues and Algorithms by Aho, Hopcroft & Ullman
4. Data Structures and Program Design by Kruse et. al., PHI
5. Data Structures using C and C++ by Tanenbaum et. al., PHI
6. Fundamentals of Data Structures in C++ by Horowitz, Sahni & Mehta
7. Data Structures in Java by Sahni
8. Algorithms + Data Structures = Programs by N. Wirth, PHI
9. How to solve it by Computers by Dromey

MCA-202 Computer Architecture [40L]

Objective:

The course Computer Architecture and Organization is being introduced in the syllabus with an objective to give a brief, lucid explanation and understanding of the general components of Computer hardware, its organization and the process through which a program interacts with the computer system.

Outcome:

After successful completion of the subject, the learners will be able to:

- Understand the organization and design of the basic Computer System and their interconnections
- Learn the way microoperations are conducted in actual through the hardware and computer programs
- Grow the concept on working procedure of Control Unit and Central Processing Unit, in general
- Categorize memory organization and explain the function of each element of a memory hierarchy.
- Identify and compare different methods for computer I/O mechanisms.

Topic Covered:

Concepts and Terminology: Digital computer components Hardware & Software and their dual nature, Role of Operating Systems (OS). **[4L]**

The ALU: ALU organization, Integer representation, Serial and Parallel Adders, is 1s and 2s complement arithmetic, Multiplication of signed binary numbers, Floating point number arithmetic, Overflow detection, Status flags. **[6L]**

Memory Unit: Memory classification, Bipolar and MOS storage cells. Organization of RAM, address decoding, Registers and stack, ROM and PROM-basic cell. Organization and erasing schemes, Magnetic memories-recording formats and methods. Disk and tape Units. Concept of memory map. Timing diagrams, T-States, Timing diagram Controlling arithmetic and logic instructions. Instruction sequencing with examples. Introduction to Micro-programming, Variations in Micro-programming configuration. **[20L]**

General Organization: Instruction work formats, Addressing modes registers, Von-Neumann concept, Interconnecting system components, Interfacing buses, Timing diagrams, Examples from popular machines. Introduction to Multiprogramming and Multiprocessing; Introduction to pipelined operation and architecture. **[10L]**

Books:

1 Hayes J. P., "Computer Architecture & Organisation", McGraw Hill,
2 Hamacher, "Computer Organisation",
3 Computer Organization and System Software, EXCEL BOOKS
4. Chaudhuri P. Pal, "Computer Organisation & Design", PHI,
5. Mano, M.M., "Computer System Architecture", PHI.
6. Burd- System Architecture,Vikas

MCA-203 Microprocessor [40L]

Objectives:

- To introduce students with the architecture and operation of typical microprocessors.
- To familiarize the students with the programming and interfacing of microprocessors.
- To provide strong foundation for designing real world applications using microprocessors.

Learning Outcomes:

At the end of this course students should:

- Understand 8085A microprocessor architecture, addressing modes and their features, Pin description, features and applications.
- Know software instruction set and assembly language programming.
- Understand interfacing memory and peripherals (IO mapped IO & Memory mapped IO).
- Understand Interrupts and DMA, Peripheral chips, A/D and D/A converters and interfacing of the same.
- Know basics of 16 bit processors and brief overview of some other microprocessors.

Topic Covered:

Introduction to microprocessors	**[2L]**
Microprocessor architecture	**[2L]**
Organisation and programming of microprocessor Intel-8085	**[8L]**
Memory interfacing	**[4L]**
Interrupt and DMA	**[4L]**
Serial and parallel communication	**[4L]**
I/O interfacing	**[2L]**
Key board & display, Programmable parallel interface, Programmable timer ADC & DAC etc	**[8L]**
Single chip microcomputer	**[2L]**
16-bit & 32-bit microprocessors	**[4L]**

Books:

1. R. Gaonkar, "Microprocessor Architecture, Programming and Applications," 5th Ed., Pearson International, 2001.
2. C. Gilmore, "Microprocessors Principles and Applications," 2nd Ed., McGraw-Hill International, 1995.
3. D. Hall, "Microprocessors and Interfacing," 2nd Ed., Tata-McGraw-Hill, 1999.
4. Liu and Gibson, "Microcomputer Systems: The 8086/8088 Family," 2nd Ed., Prentice-Hall India, 1986.
5. Treibel and Singh, "The 8088 and 8086 Microprocessors," 4th Ed., Prentice-Hall India, 1991.
6. K. Ayala, "The 8051 Microcontroller – Architecture, Programming and Applications," 2nd Ed., Pearson International, 1996.
7. Mazidi, "The 8051 Microcontrollers & Embedded Systems," Pearson Education Asia
8. M. Predco, "Programming and Customizing the 8051 Microcontroller," Tata McGraw-Hill,1999.
9. Microprocessor architecture, programming and applications with 8085/8085A, Wiley eastern Ltd, 1989 by Ramesh S. Gaonkar.
10. Intel Corp: The 8085 / 8085A. Microprocessor Book – Intel marketing communication, Wiley inter science publications, 1980.
11. An introduction to micro computers Vol. 2 – some real Microprocessor – Galgotia Book Source, New Delhi by Adam Osborne and J. Kane
12. Advanced Microprocessors by Ray and Bhurchandi - TMH

MCA-204 Numerical methods [40L]

Objective:

The objective of teaching numerical methods is to nurture the learners through an interdisciplinary approach. Through this course, the students will study the basic knowledge on the mathematical techniques that could be used by the learners in formation of algorithms as well as apply it in the areas of research.

Outcome:

Students will be able to:

- Calculate approximation, mathematical errors and its interpretation
- Apply Numerical analysis which has enormous application in the field of Science and some fields of Engineering
- Familiarize themselves with techniques such as Interpolation, Bisection Method, Differentiation, and Integration
- Understand, calculate, and interpret the Eigen values and its utility
- Solve linear and non-linear simultaneous equations.

Topic Covered:

Approximations and Errors associated with numerical methods. **[3L]**

Solution of non-linear equations: Iterative method using repeated substitutions, Bisection method, method of false position, Newton-Raphson method, Secant method, Chebyshev method. **[5L]**

Solution of linear simultaneous equations:

Direct methods: Gaussian elimination, Gauss-Jordan elimination, matrix inversion using Gauss-Jordan elimination. **[3L]**

Iterative methods: Jacobi's method, Gauss-Seidel method and their analysis. **[3L]**

Solution of non-linear simultaneous equations: Iterative method and newton-Raphson method. **[3L]**

Finding the eigen values and corresponding eigen vectors of a square matrix:

Definitions of eigenvalues and eigenvectors, Power method for finding the eigenvalues and corresponding eigenvectors of a square matrix. **[3L]**

Methods for interpolation: Newton's forward difference formula, Newton's backward difference formula, Gauss central difference formula. Divided difference formula, Lagrange's formula, iterative interpolation method. **[7L]**

Methods for differentiation: Computation of derivatives using Newton's forward/backward difference formulae. **[3L]**

Methods for integration: Trapezoidal method, Simpson's method, Boole's method, analysis and comparison of these methods, Romberg's method, Gauss quadrature formula. **[5L]**

Solution of differential equations: Euler's method, modified Euler's method, Runge-Kutta 2nd order formula, Runge-Kutta 4th order formula, predictor-corrector methods. **[5L]**

Books:

1. Numerical Analysis, Shastri, PHI
2. Numerical Analysis, S. Ali Mollah
3. Numerical Analysis, James B. Scarbarough
4. Numerical Methods for Mathematics ,Science & Engg., Mathews, PHI
5. Numerical Methods in Computer Application,Wayse,EPH

MCA-205 Software Project Management [40 L]

Learning Outcomes

- Manage the scope, cost, timing, and quality of the project, at all times focused on project success as defined by project stakeholders.
- Align the project to the organization's strategic plans and business justification throughout its lifecycle.
- Identify project goals, constraints, deliverables, performance criteria, control needs, and resource requirements in consultation with stakeholders.
- Implement project management knowledge, processes, lifecycle and the embodied concepts, tools and techniques in order to achieve project success.
- Adapt projects in response to issues that arise internally and externally.
- Interact with team and stakeholders in a professional manner, respecting differences, to ensure a collaborative project environment.
- Utilize technology tools for communication, collaboration, information management, and decision support.
- Implement general business concepts, practices, and tools to facilitate project success.
- Apply appropriate legal and ethical standards.
- Adapt project management practices to meet the needs of stakeholders from multiple sectors of the economy (i.e. consulting, government, arts, media, and charity organizations).
- Apply project management practices to the launch of new programs, initiatives, products, services, and events relative to the needs of stakeholders.
- Appraise the role of project management in organization change.

Topic Covered:

Introduction to Project management **[3L]**
Project Management Basics, Role of a Project Manager, Project Resources
Phases of Software Project

Introduction to PERT/CPM **[5L]**
Work Breakdown Structure, Network diagramming, Critical Path, PERT Probability, Crashing, Resource Leveling, Exercises,

Software Effort Estimation **[4L]**
Estimations Basics, LOC Method, Function Points, Activity Based Estimation, COCOMO Uncertainty in estimation.

Project Planning **[5 L]**
Management, Risk, Configuration, Quality Assurance, Induction, Schedule.

Configuration Management **[4L]**
Configuration Management Basics, Environment for Configuration Control , Configuration Control vs. Version Control, Code Management, Change Management, Information Management

Quality Assurance in Projects **[4L]**
Quality Basics: Quality Assurance Activities in Projects, i. Standards, ii. Coding Standards, iii. Documentation Standards, iv. Design Standards, v. Templates, vi. Formats, vii. Processes
Quality Control Activities in Projects, i. Verification, ii. Validation, iii. Quality Metrics
Introduction to ISO 9000, SEI – CMM Maturity Levels, Six Sigma

Productivity Aspects **[2L]**
Productivity Basics, Productivity Measurement & Metrics

Human Factors and Leadership **[4L]**

Motivation, Communication, Handling Difficult People, Leadership, Team Dynamics

Progress Tracking & Control **[3L]**

Progress Assessment & Reporting, Scope Management, Risk Mitigation

Project Closeout **[3L]**

Project post-mortem, Collection of re-usable Components, Draw lessons from the good & bad Practices of the project , Project-End Audit

Organizational Support for Effective Project Management **[3L]**

Recognition as a Specialist Discipline, Organize Knowledge Repository, Processes, Standards & Guidelines, Training

Books:

1. Gilb, T., "Principles of Software Engineering Management", Addison Weskey. Reading. M.A 1988.
2. Putnam. L.H., Myers. W., "Industrial Sire: Software - Effective Management using Measurement". IEEE C.S. Press. 1997.

MCA-291 Data Structure Lab

Objective: The main aim of data structure lab is to implement the theoretical concepts of different topics of data structure into practically through programs. The students will grow more interest and deep knowledge of different items in linear and non-linear data structure.

Subject outcome:

- The students can implement stack and queue using array and linked list in programmatically.
- They will able to perform different operations of stack and queue.
- Student can implement sparse matrix.
- They will able to write programs for implementing polynomial and their operations.
- They can implement the traversal of binary tree.
- They will able to write programs for different sorting and searching algorithms.

Topic Covered:

Stacks and Queues: adding, deleting elements Circular Queue: Adding & deleting elements Merging Problem : Evaluation of expressions operations on Multiple stacks & queues :
Implementation of linked lists: inserting, deleting, inverting a linked list.
Implementation of stacks & queues using linked lists: Polynomial addition, Polynomial multiplication Sparse Matrices : Multiplication, addition.
Recursive and Nonrecursive traversal of Trees
Threaded binary tree traversal.
AVL tree implementation Application of Trees.
Application of sorting and searching algorithms

Hash tables implementation: searching, inserting and deleting, searching & sorting techniques.

List of Experiments: Experiments should include but not limited to :

1. Write a program in C to implement simple Stack, Queue, Circular Queue, Priority Queue.
2. Write a menu driven program that implements singly linked list for the following operations: Create, Display, Concate, merge, union, intersection
3. Write a menu driven program that implements doubly linked list for the following operations: Create, Display, Count, Insert, Delete, Search, Copy, Reverse, Sort
4. Write a menu driven program that implements doubly linked list for the following operations: Create, Display, Concate, merge, union, intersection
5. Write a menu driven program that implements Singly circular linked list for the following operations: Create, Display, Count, Insert, Delete, Search, Copy, Reverse, Sort
6. Write a program in C for sorting methods.
7. Write a menu driven program in C to Create a binary search tree, Traverse the tree in Inorder, Preorder and Post Order , Search the tree for a given node and delete the node
8. Write a program in C to implement insertion and deletion in B tree.
9. Write a program in C to implement insertion and deletion in AVL tree ,
10. Write a menu driven program that implements Heap tree (Maximum and Minimum Heap tree) for the following operations. (Using array) Insert, Delete .
11. Write a program to implement double hashing technique to map given key to the address space. Also write code for collision resolution (linear probing).
12. Write a program in C to implement Dijkstra's shortest path algorithm for a given directed graph.
13. Write a program in C to insert and delete nodes in graph using adjacency matrix.
14. Write a program in C to implement Breadth First search using linked representation of graph.
15. Write a program in C to implement Depth first search using linked representation of graph.
16. Write a program in C to create a minimum spanning tree using Kruskal's algorithm.
17. Write a program in C to create a minimum spanning tree using Prim's algorithm, etc.

Books:

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., New Delhi.
2. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education Asia, Delhi-2002
3. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., New Delhi.
4. K Loudon, "Mastering Algorithms with C", Shroff Publisher & Distributors Pvt. Ltd.
5. Bruno R Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", Jhon Wiley & Sons, Inc.
6. Adam Drozdek, "Data Structures and Algorithms in C++", Thomson Asia Pvt. Ltd.(Singapore)

MCA-292 Microprocessor Lab

Objectives:

- Learning of 8085 microprocessor programming.
- Learn the design aspects of I/O and Memory Interfacing circuits.

Learning Outcomes:

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

- Work with 8085 trainer kit components, including the memory map.
- Design and implement programs on 8085 microprocessor.
- Design interfacing circuits with 8085

Topic Covered:

Experiments should include but not limited to :

1. Familiarization with 8085 register level architecture and trainer kit components, including the memory map.
2. Familiarization with the process of storing and viewing the contents of memory as well as registers.
3. Study of prewritten programs on trainer kit using the basic instruction set (data transfer, Load/Store, Arithmetic, Logical)
4. Familiarization with 8085 simulator on PC.
5. Study of prewritten programs using basic instruction set (data transfer, Load/Store, Arithmetic, Logical) on the simulator.
6. Programming using kit/simulator for i) look up table, ii) Copying a block of memory iii) Shifting a block of memory, iv) Packing and unpacking of BCD numbers , v) Addition of BCD numbers , vi) Binary to ASCII conversion , vii) String Matching , viii) Multiplication using Booth's Algorithm
7. Program using subroutine calls and IN/OUT instructions using 8255 PPI on the trainer kit eg, subroutine for delay, reading switch state & glowing LEDs accordingly, finding out the frequency of a pulse train etc.
8. Interfacing any 8-bit Latch (eg, 74LS373) with trainer kit as a peripheral mapped output port with absolute address decoding
9. Interfacing with I/O modules: ADC, Speed control of mini DC motor using DAC , Keyboard, Multi-digit Display with multiplexing, Stepper motor.
10. Writing programs for 'Wait Loop (busy waiting)' and ISR for vectored interrupts (eg, counting number of pulses within specified time period).
11. Study of 8051 Micro controller kit and writing programs for the following tasks using the kit, Table look up, Basic arithmetic and logical operations, Interfacing of Keyboard and stepper motor.
12. Familiarization with EPROM programming and Erasing

Books:

1. Microprocessor architecture, programming and applications with 8085/8085A, Wiley eastern Ltd, 1989 by Ramesh S. Gaonkar.
2. Intel Corp: The 8085 / 8085A. Microprocessor Book – Intel marketing communication, Wiley inter science publications, 1980.
3. An introduction to micro computers Vol. 2 – some real Microprocessor – Galgotia Book Source, New Delhi by Adam Osborne and J. Kane
4. Advanced Microprocessors by Ray and Bhurchandi - TMH
5. Intel Corp. Micro Controller Handbook – Intel Publications, 1994.
6. Microprocessors and Interfacing by Douglas V. Hall, McGraw Hill International Ed. 1992
7. Assembly Language Programming the IBM PC by Alan R. Miller, Subex Inc, 1987
8. The Intel Microprocessors: 8086/8088, 80186, 80286, 80386 & 80486, Bary B. Brey, Prentice Hall, India 1996.

MCA-293 Numerical Lab

Experiments should include but not limited to :

1. Write a program to find the determinant of a 3X3 matrix.
2. Write a program to implement upper triangular matrix and using it find determinant.
3. Write a program to find the determinant of a n x n matrix where n>3.
4. Write a program to find the solution of a system of linear equations using Gauss elimination method.
5. Write a program to find the solution of a system of linear equations using Matrix inversion method.
6. Write a program to find the solution of a system of linear equations using Gauss-Seidal method.
7. Write a program to implement Newton's forward interpolation method.
8. Write a program to implement Newton's backward interpolation method.
9. Write a program to implement Lagrange's interpolation method.
10. Write a program to implement Newton's Divided Difference Interpolation method.
11. Write a program to find the root using bisection method.
12. Write a program to find the root using Regula-falsi method.
13. Write a program to find the root using Newton's Raphson method.
14. Write a program to implement Trapezoidal rule.
15. Write a program to implement Simpson's 1/3 rule.
16. Write a program to implement Simpson's 3/8 rule.
17. Write a program to implement Weddle's rule.
18. Write a program to implement Runge-Kutta method of order 2.
19. Write a program to implement Runge-Kutta method of order 4.
20. Write a program to implement Euler's method.
21. Write a program to implement Modified Euler's method.

SECOND YEAR FIRST SEMESTER (Semester-III)

MCA-301 Database Management System (DBMS) [40 L]

Objective: This course is intended to provide you with an understanding of the current theory and practice of database management systems. To help you more fully appreciate their nature, the course provides a solid technical overview of database management systems, using a current database product as a case study. In addition to technical concerns, more general issues are emphasized. These include data independence, integrity, security, recovery, performance, database design principles, and database administration.

Outcome:

At the completion of this course, students should be able to do thefollowing:

- Understand the role of a database management system in an organization.
- Understand basic database concepts, including the structure and operation of the relational data model.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).
- Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
- Design and implement a small database project using SQL.
- Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.
- Describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.
- Understand the role of the database administrator.

Topic Covered:

Introduction: Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Three Schema architecture of DBMS. **[3L]**

Entity-Relationship Model: Basic concepts, Design Issues, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features. **[5L]**

Relational Model: Structure of relational Databases, Relational Algebra, Relational Calculus, Extended Relational Algebra Operations, Views, Modifications Of the Database. **[6L]**

SQL and Integrity Constraints :

Concept of DDL, DML, DCL. Basic Structure, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Subqueries, Database security application development using SQL, Stored procedures and triggers. **[6 L]**

Relational Database Design : Functional Dependency, Different anamolies in designing a Database., Normalization using funtional dependencies, Decomposition, Boyce-Codd Normal Form, 3NF, Nomalization using multi-valued depedencies, 4NF, 5NF **[7 L]**

Internals of RDBMS: Physical data structures, Query optimization : join algorithm, statistics and cost bas optimization. Transaction processing, Concurrency control and Recovery Management : transaction model properties, state serializability, lock base protocols, two phase locking. Database Security **[5 L]**

File Organization & Index Structures: File & Record Concept, Placing file records on Disk, Fixed and Variable sized Records, Types of Single-Level Index (primary, secondary, clustering), Multilevel Indexes, Dynamic Multilevel Indexes using B tree and B+ tree . **[6 L]**

Advanced Concepts: Object-oriented database concepts and other query languages **[2L]**

Books:

1. Henry F. Korth and Silberschatz Abraham, "Database System Concepts", Mc.Graw Hill.
2. Elmasri Ramez and Novathe Shamkant, "Fundamentals of Database Systems", Benjamin Cummings Publishing. Company.
3. Ramakrishnan: Database Management System , McGraw-Hill
4. Gray Jim and Reuter Address, "Transaction Processing : Concepts and Techniques", Moragan Kauffman Publishers.
5. Jain: Advanced Database Management System CyberTech
6. Date C. J., "Introduction to Database Management", Vol. I, II, III, Addison Wesley.
7. Ullman JD., "Principles of Database Systems", Galgottia Publication.
8. James Martin, "Principles of Database Management Systems", 1985, Prentice Hall of India, New Delhi
9. "Fundamentals of Database Systems", Ramez Elmasri, Shamkant B.Navathe, Addison Wesley Publishing Edition
10. "Database Management Systems", Arun K.Majumdar, Pritimay Bhattacharya, Tata McGraw Hill

MCA-302 Theory of Computing [40 L]

Objective: Course should provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (and less magical) view towards algorithmic design and in general computation itself. The course should in addition clarify the practical view towards the applications of these ideas in the theoretical computer science.

Outcome: After completing the course, the student will be able to:

- Model, compare and analyse different computational models using combinatorial methods.
- Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
- Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.
- Identify limitations of some computational models and possible methods of proving them.
- Have an overview of how the theoretical study in this course is applicable to and engineering application like designing the compilers.

Detail Syllabus:

Finite State Machines : Definition, concept of sequential circuits, state table & state assignments, concept of synchronous, asynchronous and liner sequential machines. **[3L]**

Finite State Models : Basic definition, mathematical representation, Moore versus Mealy m/c, capability & limitations of FSM, state equivalence & minimization, machine equivalence, incompletely specified machines, merger graph & compatibility graph, merger

table, Finite memory, definite, information loss less & inverse machines : testing table & testing graph. **[6 L]**

Structure of Sequential Machines : Concept of partitions, closed partitions, lattice of closed partitions, decomposition : serial & parallel. **[4L]**

Finite Automation : Preliminaries (strings, alphabets & languages, graphs & trees, set & relations), definition, recognition of a language by an automata - idea of grammar, DFA, NFA, equivalence of DFA and NFA, NFA with e-moves, regular sets & regular expressions : equivalence with finite automata, NFA from regular expressions, regular expressions from DFA, two way finite automata equivalence with one way, equivalence of Moore & Mealy machines, applications of finite automata. **[5L]**

Closure Properties of Regular Sets : Pumping lemma & its application, closure properties minimization of finite automata : minimization by distinguishable pair, Myhill-Nerode theorem. **[4L]**

Context Free Grammars : Introduction, definition, derivation trees, simplification, CNF & GNF. **[5L]**

Pushdown Automata : Definition, moves, Instantaneous Descriptions, language recognised by PDA, deterministic PDA, acceptance by final state & empty stack, equivalence of PDA and CFL. **[6L]**

Closure Properties of CFLs : Pumping lemma & its applications, ogden's lemma, closure properties, decision algorithms. **[4L]**

Introduction to Z. Regular language properties and their grammars. Context sensitive languages. **[3L]**

Books :

1. Hopcroft JE. and Ullman JD., "Introduction to Automata Theory, Languages & Computation", Narosa.
2. K.L.P Mishra & N. Chandrasekharan – "Theory of Computer Science", PHI
3. Ash & Ash – "Discrete Mathematics",TMH
4. Martin—Introduction
5. Lewis H. R. and Papadimitrou C. H., "Elements of the theory of Computation", P.H.I.
6. Kain, "Theory of Automata & Formal Language", McGraw Hill.
7. Kohavi ZVI, "Switching & Finite Automata", 2nd Edn., Tata McGraw Hill.
8. Linz Peter, "An Introduction to Formal Languages and Automata", Narosa
9. "Introduction to Formal Languages", Tata McGraw Hill, 1983.

MCA-303 Design and Analysis of Algorithm [40 L]

Course Objectives

Upon completion of this course, students will be able to do the following:

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

Course Outcomes

Students who complete the course will have demonstrated the ability to do the following:

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- 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.
- 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.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.
- 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.
- Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.
- Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.
- Explain what amortized running time is and what it is good for. Describe the different methods of amortized analysis (aggregate analysis, accounting, potential method). Perform amortized analysis.
- Explain what competitive analysis is and to which situations it applies. Perform competitive analysis.
- Compare between different data structures. Pick an appropriate data structure for a design situation.
- Explain what an approximation algorithm is, and the benefit of using approximation algorithms. Be familiar with some approximation algorithms, including algorithms that are PTAS or FPTAS. Analyze the approximation factor of an algorithm.

Topic Covered:

Models of computation : RAM,TM etc. time and space complexity **[2L]**

Asymptotic Notation: Big-O, omega, theta etc.; finding time complexity of well known algorithms like- heapsort, search algorithm etc. **[3L]**

Algorithm Design techniques: Recursion- Definition, Use, Limitations, Examples: Hanoi problem. Tail Recursion **[3L]**

Divide and Conquer : Basic method, use, Examples: Merge sort, Quick Sort, Binary Search **[3L]**

Dynamic Programming: Basic method, use, Examples: matrix-chain multiplication, All pair shortest paths, single-source shortest path, Travelling Salesman problem **[3L]**

Branch and Bound : Basic method, use, Examples: The 15-puzzle problem **[3L]**

Backtracking: Basic method, use, Examples: Eight queens problem, Graph coloring problem, Hamiltonian problem **3L]**

Greedy Method : Basic method, use, Examples: Knapsack problem, Job sequencing with deadlines, minimum spanning tree(Prim's and Kruskal's algorithms) **[3L]**

Lower Bound Theory : Bounds on sorting and sorting techniques using partial and total orders. **[3L]**

Disjoint Set Manipulation : Set manipulation algorithm like UNION-FIND, union by rank, Path compression. **[2L]**

Properties of graphs and graph traversal algorithms: BFS and DFS **[3L]**

Matrix manipulation algorithms :Different types of algorithms and solution of simultaneous equations, DFT & FFT algorithm; integer multiplication schemes **[3L]**

Notion of NP-completeness : P class, NP-hard class, NP-complete class, Circuit Satisfiability problem, Clique Decision Problem. **[3L]**

Approximation algorithms : Necessity of approximation scheme, performance guarantee, Polynomial time approximation schemes: 0/1 knapsack problem. **[3L]**

Books:

1. A.Aho, J.Hopcroft and J.Ullman "The Design and Analysis of algorithms"
2. D.E.Knuth "The Art of Computer Programming", Vol. I & Vol.2
3. Horowitz Ellis, Sahani Sartaz, R. Sanguthevar " Fundamentals of Computer Algorithms".
4. Goodman: Introduction to Design and Analysis Of Algorithms TMH
1. K.Mehlhorn, "Data Structures and algorithms- Vol. I & Vol. 2
2. E.Horowitz and Shani "Fundamentals of Computer algorithms"
3. E.M.Reingold, J.Nievergelt and N.Deo- "Combinational algorithms- Theory and Practice", Prentice Hall , 1997
4. A.Borodin and I.Munro, "The computational complexity of Algebraic and Numeric problems"

MCA-304 Object Oriented Programming [40 L]

Objective: The primary objective of object oriented programming is to understand the real world problems into objects, inheritance, classes, hiding, binding etc. The students will be able how to reuse of code, how to model a problem for OOP.

Subject outcome:

- The students will know the difference between procedure oriented and object oriented; top down vs. bottom up approach.
- Students will learn class, object, data abstraction, data hiding, data encapsulation, polymorphism, message passing etc.
- They will know the concept of constructor and destructor and friend function.
- They will able to learn function overloading, operator overloading, and virtual function.
- The learner will able to learn details in different types of inheritance.
- They will able to handle file manipulation functions.
- They will learn template and exception handling.

Topic Covered:

Introduction: Why object orientation, History and development of Object Oriented Programming language, concepts of object oriented programming language **[3L]**
Basic concepts of object oriented programming using Java: Object, class, message passing, encapsulation, and polymorphism, difference between OOP and other conventional programming-advantages and disadvantages. **[3L]**
Classes and Objects: defining classes, defining member functions, access specifiers (i.e. Private, public, protected), array of objects, objects as function arguments, friend function, function returning objects, pointers to members. **[4L]**
Constructor and Destructor: constructors, parameterized constructors, multiple constructors in class, dynamic initialization of objects, destructors. **[3L]**
Memory Management and pointers: Using new operator, comparison of new over malloc, calloc and realloc, using delete operator. **[4L]**
Pointer, Virtual Functions, Polymorphism: pointers, pointers to objects, this pointer, pointers to derived class object, virtual functions, pure virtual functions etc. **[3L]**
Inheritance: Extending class, types of inheritance, use of virtual base class, abstract class, constructors in derived class. **[3L]**
Operator Overloading: defining operator overloading, overloading - (unary, binary operators), overloading operators using friends, rules for overloading operators. **[4L]**
Working I/O and files: Streams, unformatted and formatted I/O operations, managing output with manipulators. Creating/ opening / closing / deleting files, file pointers, random access to file, command line arguments. **[4L]**
Templates: class templates, function templates, overloading of template function. **[3L]**
Exception Handling: Throwing, catching, re-throwing an exception, specifying exceptions etc. **[3L]**
STL (Standard template library): components of STL, containers, algorithms, iterators, function objects. **[3L]**

Books :

1. Ali Bahrami, - “Object –Oriented System Development” - Mc Graw Hill.
2. Rambaugh, James Michael, Blaha - “Object Oriented Modelling and Design” - Prentice Hall India/ Pearson Education
3. Bruce, Foundations of Object Oriented Languages, PHI
4. Patrick Naughton, Herbert Schildt – “The complete reference-Java2” - TMH
5. Priestley – “ Practical Object Oriented Design using UML” - TMH
6. Jana, C++ & Object Oriented Programming, PHI
7. Alhir, learning UML, SPD/O’Reily
8. Page Jones, Meiler - “Fundamentals of object oriented design in UML”
9. Roff: UML: A Beginner’s Guide TMH
10. Rajaram: Object Oriented Programming and C++, New Age International
11. Mahapatra: Introduction to System Dynamic Modelling, Universities Press
12. Muller : Instant UML, Shroff Publishers / Wrox
13. Srimathi, Object Oriented Analysis & Design Using UML, Scitech

MCA-305 Software Engineering [40 L]

Objective:

- Knowledge of basic SW engineering methods and practices, and their appropriate application.
- Describe software engineering layered technology and Process frame work.
- A general understanding of software process models such as the waterfall and evolutionary models.
- Understanding of software requirements and the SRS documents.
- Understanding of the role of project management including planning, scheduling, risk management, etc.
- Describe data models, object models, context models and behavioural models.
- Understanding of different software architectural styles.
- Understanding of implementation issues such as modularity and coding standards.
- Understanding of approaches to verification and validation including static analysis, and reviews. 10. Understanding of software testing approaches such as unit testing and integration testing.
- Describe software measurement and software risks.
- Understanding of software evolution and related issues such as version management.
- Understanding on quality control and how to ensure good quality software.

Outcome:

- Basic knowledge and understanding of the analysis and design of complex systems.
- Ability to apply software engineering principles and techniques.
- Ability to develop, maintain and evaluate large-scale software systems.
- To produce efficient, reliable, robust and cost-effective software solutions.
- Ability to perform independent research and analysis.
- To communicate and coordinate competently by listening, speaking, reading and writing english for
- technical and general purposes.
- Ability to work as an effective member or leader of software engineering teams.
- To manage time, processes and resources effectively by prioritising competing demands to achieve
- personal and team goals Identify and analyzes the common threats in each domain.
- Ability to understand and meet ethical standards and legal responsibilities.

Topic Covered:

UNIT – 1: The Product : Software, Software Myths, The process : Software engineering : A Layered Technology, Software Process Models, The linear sequential Model, The prototyping Model, The RAD Model, Evolutionary Software Process Models, Component – Based Development, Fourth Generation Techniques, Software process and project metrics : Software measurement. **[6L]**

UNIT – 2: Software Design : Problem Partitioning, Top-Down And Bottom-Up design, Decision tree, decision table, Cohesion, Coupling, Design approaches : Functional and Object- Oriented approach.

Function- Oriented Software Design : SA/SD methodology, Structure analysis, DFD, Structure chart, Design review
Object-Oriented Software Design : Concept of OO Software – Design and Analysis, Overview of various UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns. **[8L]**

UNIT – 3 Coding : Coding standards, code walk- throughs, code inspection, clean room testing and documentation. **[6L]**

UNIT – 4 Software Quality Assurance : Quality Concepts, The Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, Mistake Proofing for Software, Introduction to ISO standard. **[10L]**

UNIT – 5 Software Testing Technique : Software testing fundamentals, Test case design, White-box Testing, Basis path testing, Control structure testing, Black-box testing, Testing for specialized environments, architectures and application. **[5L]**
UNIT – 6 Software Maintenance: Characteristics, reverse engineering, maintenance process models, estimation of cost. **[8L]**

Books:
1. Bruegge, Bernd and Allen H. Dutoit. "Object-Oriented Software Engineering: Using UML, Patterns and Java", Pearson: Prentice Hall Publishers.
2. Braude, E. J. "Software Engineering: An Object-Oriented Perspective". Wiley.
3. Schmuller, Joseph. "SAMS Teach Yourself UML in 24 Hours". Sams Publishing.
4. Sommerville, Ian. "Software Engineering". Addison-Wesley.

MCA-391 DBMS Lab

Experiments should include but not limited to :

Structured Query Language

1. Creating Database

- Creating a Database
- Creating a Table
- Specifying Relational Data Types
- Specifying Constraints
- Creating Indexes

2. Table and Record Handling

- INSERT statement
- Using SELECT and INSERT together
- DELETE, UPDATE, TRUNCATE statements
- DROP, ALTER statements 3. Retrieving Data from a Database
- The SELECT statement

- Using the WHERE clause
- Using Logical Operators in the WHERE clause
- Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING

3. Clause

- Using Aggregate Functions
- Combining Tables Using JOINS
- Subqueries 4. Database Management
- Creating Views Creating Column Aliases
- Creating Database Users
- Using GRANT and REVOKE

Cursors in Oracle PL / SQL

Writing Oracle PL / SQL Stored Procedures.

Use of user interfaces and report generation utilities typically available with RDBMS products.

MCA-392 OOPs Lab

Objective: The primary objective of object oriented programming is how to implement theoretical concepts of OOP into practical. The students will gain more knowledge in different topics of OOP and attain their confidence high.

Subject outcome:

- The students will able to implement simple programs through class and object.
- The learner can implement different types of constructor, also can develop for destructor.
- Can implement overloading and overriding.
- Can implement virtual function and abstract class.
- They can implement real problems with related to inheritance.
- They can implement file related operations.
- They will able to implement template class and function.

Topic Covered:

Experiments should include but not limited to :

1. Assignments on class, constructor, parameterized constructor.
2. Assignments on function overloading and overriding.
3. Assignments on pointers, virtual function, abstract class.

4. Assignments on operator overloading, friend function.
5. Assignments on function template and class template.
6. Assignments on file handling.
7. Assignments on exception handling and STL.

MCA-393 Term Paper-I

Seminar topic will be assigned to individual student by the Head of the department at the beginning of the semester.

MCA-394 Industrial Visit

An Industrial Visit would be organized by the department for not less than 3 days and not more than one week and students should submit a report on that tour which will be examined by a board of examiners to be nominated by the B.O.S.

SECOND YEAR SECOND SEMESTER (Semester-IV)

MCA-401 Computer Graphics [40L]

Objective:

Students will try to learn:

- To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
- To learn the basic principles of 3- dimensional computer graphics.
- Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.
- Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.
- To be able to discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications.

Outcome:

Students will able to:

- To list the basic concepts used in computer graphics.
- To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
- To describe the importance of viewing and projections.

Topic Covered:

Introduction to computer graphics & graphics systems [3L]

Overview of computer graphics, representing pictures, preparing, presenting & interacting with pictures for presentations; Visualization & image processing; RGB color model, direct coding, lookup table; storage tube graphics display, Raster scan display, 3D viewing devices, Plotters, printers, digitizers, Light pens etc.; Active & Passive graphics devices; Computer graphics software.

Scan conversion: [7L]

Points & lines, Line drawing algorithms; DDA algorithm, Bresenham's line algorithm, Circle generation algorithm; Ellipse generating algorithm; scan line polygon, fill algorithm, boundary fill algorithm, flood fill algorithm.

2D transformation & viewing [10L]

Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, transformations between coordinate systems; reflection shear;

Transformation of points, lines, parallel lines, intersecting lines. Viewing pipeline, Window to viewport co-ordinate transformation, clipping operations, point clipping, line clipping, clipping circles, polygons & ellipse.

3D transformation & viewing **[10L]**

3D transformations: translation, rotation, scaling & other transformations. Rotation about an arbitrary axis in space, reflection through an arbitrary plane; general parallel projection transformation; clipping, viewport clipping, 3D viewing.

Curves **[4L]**

Curve representation, surfaces,designs, Bezier curves, B-spline curves, end conditions for periodic B-spline curves, rational B-spline curves.

Hidden surfaces **[3L]**

Depth comparison, Z-buffer algorithm, Back face detection, BSP tree method, the Printer's algorithm, scan-line algorithm; Hidden line elimination, wire frame methods, fractal - geometry.

Color & shading models **[3L]**

Light & color model; interpolative shading model; Texture;

Book:

1. Procedural Elements of Computer Graphics: D.E.Rogers, McGraw Hill.
2. Computer Graphics: Hearn and Baker, Prentice Hall India.
3. Computer Graphics, principles & practices by J.D. Foley, A. van Dam, S.K. Feiner and J.F. Huges, Addison Wesley.
4. Computer Graphics, a programming approach, by S. Harrington, TMH publication.
5. A.N.sinha, A.D.Udai, "Computer Graphics", TMH, New Delhi,2008.
6. M.K.Pakhira, "Computer Graphics, Multimedia and Animation" PHI Pvt. Ltd., 2008.
7. Foley, van Dam, Feiner, Hughes, " Computer Graphics : Principles and Practice", Addison Wesley.

MCA-402 Operating System [40L]

Objective:

It is hard for the computers to work with ease without an operating system. Therefore, every computers are provides with some type of operating system to coordinate between the users and hardware of the system. The objective is to grow a clear concept regarding the working of the operating system, its policies, scheduling, synchronization, and memory storage management.

Outcome:

Through this course, students will be able to:

- Understand the role, objectives, and policies with which an operating system works in a computer system
- Understand the process management policies and scheduling of processes by CPU
- Gain knowledge to synchronize process and its coordination with all the processes
- Describe and analyze the memory management and its allocation policies.

- Identify use and evaluate the storage management policies with respect to different storage management technologies

Topic Covered:

Introduction to Operating Systems [**1L]**

Concept of batch-processing, multi-programming, time sharing, real time operations [**2L]**

Process Management: Concept of process, state diagram, process control block; scheduling of processes – criteria, types of scheduling, non-preemptive and preemptive scheduling algorithms like: FCFS, Shortest Job First/Next (SJF/N), Shortest Remaining Time Next (SRTN), Round Robin (RR), Highest Response ratio Next (HRN), Priority based scheduling, different Multilevel queue scheduling etc.; [**5L]**

Threads – concept, process vs thread, kernel and user threads, multithreading models [**2L]**

Inter-process Communication (IPC) – Shared memory, message, FIFO, concept of semaphore, critical region, monitor [**2L]**

Process Synchronization: concepts, race condition, critical section problem and its solutions; synchronization tools- semaphore, monitor etc., discussion of synchronization problems like producer-consumer, readers-writers, dining philosophers, sleeping-barber etc.

Deadlock – conditions, resource allocation graph, prevention techniques, avoidance technique – Banker's algorithm and related algorithms [**6L]**

Memory management: Address space and address translation; static partitioning, dynamic partitioning, different types of fragmentation, paging, segmentation, swapping, virtual memory, demand paging, page size, page table, page replacement algorithms – FIFO, LRU, Optimal page replacement, Variants of LRU, etc; thrashing, working set strategy **[6L]**

File Management: File and operations on it, file organization and access; file allocation; directory structures, file sharing, file protection [**4L]**

Device management: Magnetic disks, disk scheduling- criteria, algorithms – FCFS, SSTF, SCAN, C-SCAN, LOOK, etc, disk management – formatting, boot block, disk free space management techniques, concept of RAID etc [**3L]**

Protection and Security: Concepts of domain, Access matrix and its implementation, access control, Security of systems- concepts, threats- Trojan horse, virus, worms etc, introduction to cryptography as security tool, user authentication [**5L]**

Case Studies [**4L]**

Books:

1. Operating Systems Concepts – A. Silberschatz, P. Galvin and G. Gagne. Wiley India
2. Operating Systems Concepts - Gary Nutt, N. Chaki and S. Neogy, Pearson Education
3. Operating Systems – W. Stallings, Pearson Education
4. Operating Systems: A Concept-based Approach – D. M. Dhamdhere, Tata McGraw-Hill

MCA-403 Computer Network [40L]

Objective: The main emphasis of this section is on the organization and management of local area networks (LANs). The course objectives include learning about computer network organization and implementation, obtaining a theoretical understanding of data communication and computer networks, and gaining practical experience in installation, monitoring, and troubleshooting of current LAN systems. The course introduces computer

communication network design and its operations. The course includes the following topics: Open Systems Interconnection (OSI) communication model; error detection and recovery; local area networks; bridges, routers and gateways; network naming and addressing; and local and remote procedures. On completion of the course, the student should be able in part to design, implement and maintain a typical computer network (LAN).

Outcome:

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

- Describe the general principles of data communication.
- Describe how computer networks are organized with the concept of layered approach.
- Describe how signals are used to transfer data between nodes.
- Implement a simple LAN with hubs, bridges and switches.
- Describe how packets in the Internet are delivered.
- Analyze the contents in a given Data Link layer packet, based on the layer concept.
- Design logical sub-address blocks with a given address block.
- Decide routing entries given a simple example of network topology
- Describe what classless addressing scheme is.
- Describe how routing protocols work.

Topic Covered:

Introduction: Uses of Computer Networks, Types of Computer Networks, OSI Reference Model, Example Networks **[4L]**

Physical Layer: Data and signal fundamentals, Transmission impairments, Attenuation, Distortion, Noise, Data rate limits for noisy and noiseless channels, Performance **[2L]**

Digital Transmission – Problems with digital transmission, Different line coding schemes, Block coding schemes, Scrambling techniques, Analog to digital encoding. Analog Transmission. **[4L]**

Transmission Media - Guided (wired) media – Twisted pair cable, Coaxial cable and Fibre optic cable, Unguided (wireless) media – Different propagation modes, Radio waves, Terrestrial microwaves, Satellite communication. **[2L]**

Concept of multiplexing, Frequency division multiplexing, Time division multiplexing – Synchronous and Statistical time division multiplexing, Handling variable length data, Pulse stuffing. Concept of spreading spectrum, Frequency hopping spread spectrum and Direct sequence spread spectrum. **[4L]**

Data Link Layer: Link Layer Services, Error detection and Correction Techniques, Multi Access Protocols, Link Layer Addressing, Ethernet, Hubs, Switches and Switches, Point to Point Protocol, Asynchronous Transfer Mode, Multiprotocol Label Switching **[6L]**

Network Layer: Introduction, Virtual Circuit and Datagram Networks, IP Addressing, Subnetting, Routing Algorithms (Link State, Distance Vector, Hierarchical), Routing in the Internet (RIP, OSPF, BGP), Broadcast and Multicast Routing Algorithms, Routers,ICMP, IPv6 **[8L]**

Transport Layer: Introduction to Transport Layer Services, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Connection Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control, Sockets, Quality of services (QOS) **[6L]**

Application Layer: Web and HTTP, Domain Name Space (DNS), Electronic Mail (SMTP, MIME, IMAP, POP3), File Transfer Protocol, Cryptography **[4L]**

Books:

1. **Computer Networking: A Top-Down Approach Featuring the Internet**, by James F. Kurose and Keith W. Ross, 5th Edition, Pearson Education, 2010

2. **Data communication and Networking**, by Behrouz A. Forouzan, 4th Edition, Tata McGraw-Hill, 2007

3. **Computer Networks,** by Andrew S. Tanenbaum, 4th Edition, Prentice Hall India, 2003

4. **Computer Networks: A Systems Approach**, by Larry L. Peterson and Peter S. Davie,4th Edition, Morgan Kauffman Publishers, 2007
5. **Data and Computer Communication**, by William Stallings, 9th Edition, Pearson Education, 2011
6. **Unix Network Programming: Networking APIs: Sockets and XTI, (Volume 1)** by W. Richard Stevens, 2nd Edition, Prentice Hall India, 1999

MCA-404 Compiler Design [40L]

Objective:

The main objective of this course is to introduce the major concept areas of language translation and compiler design and to develop an awareness of the function and complexity of modern compilers. This course is a study of the theory and practice required for the design and implementation of interpreters and compilers for programming languages.

Outcome:

By the end of the course, the successful student will be able to do:

- To realize basics of compiler design and apply for real time applications.
- To introduce different translation languages
- To understand the importance of code optimization
- To know about compiler generation tools and techniques
- To learn working of compiler and non compiler applications
- Design a compiler for a simple programming language
- To convert from source language to target language and should be recognize what happens at each and every phase of a compiler.
- To understand the different types of parsing techniques and should be in a position to solve the problem.
- To write the code by using YACC and lex.

Topic Covered:

Introduction to system softwares **[1L]**
Assembler: functions, features, design of one pass and two pass assemblers **[5L]**
Macroprocessors: functions, features and design **[2L]**
Introduction to Compiler, Different phases and passes of compiler **[1L]**
Lexical Analysis: Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Finite state machines and regular expressions and their applications to lexical analysis, Implementation of lexical analyzers **[4L]**
Syntax Analysis: Role of the parser, Formal grammars and their application to syntax analysis, Context free grammars, Derivation and parse trees, Top Down parsing, LL(1) grammars, Predictive Parsing, Bottom-up-parsing, Shift Reduce Parsing, LR(0) grammars, LR parsing algorithms **[10L]**
Syntax Directed Translation: Syntax directed definitions, Construction of syntax trees, Bottom-up evaluation of S-attributed definitions, L-attributed definitions **[2L]**
Runtime Environments: Source Language issues, Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter passing mechanism **[2L]**

Intermediate Code Generation: Intermediate languages, Graphical representation, Threeaddress code, Implementation of three address statements (Quadruples, Triples, Indirect triples) **[2L]**

Code Optimization and generation: Introduction, Basic blocks and flow graphs, Transformation of basic blocks, DAG representation of basic blocks, Principle sources of optimization, Loops in flow graph, Peephole optimization. Issues in the design of code generator, Register allocation and assignment **[3L]**

Loader and Linkers: Basic Concepts of Linkers and Loader Functions, Boot Loaders, Linking Loaders, Linkage Editors, Dynamic Linking **[5L]**

Concept of Editor and text editor, Interpreters, Simulator, Text editors - Overview of the Editing Process - User Interface – Editor Structure. – Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User-Interface Criteria. **[3L]**

Books:

1. Alfred Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman, "Compilers Principles, Techniques and Tools", Pearson Education Asia (2nd Ed. - 2009).
7. Leland L. Beck**,** "System Software: An Introduction to Systems Programming", 3/E, Addison-Wesley, 1997.
3. Allen I. Holub "Compiler Design in C", Prentice Hall of India, 2003.
4. C. N. Fischer and R. J. LeBlanc, "Crafting a compiler with C", Pearson Education.
5. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2003.
6. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
7. Kenneth C. Louden, "Compiler Construction: Principles and Practice", Thomson Learning.
8. Systems Programming and Operating Systems – D. M. Dhamdhere, TMH
9. John J. Donovan, " Systems Programming", 3rd edition, 1997, Addison Wesley.

MCA-405 Optimization Technique [40L]

Objective:

In every sphere of life, people deal with optimization. Optimization techniques help to use the available scarce resources to its optimality. The objective is to study the basic components of an optimization problem and formulation of design problems as mathematical programming problems.

Outcome:

The outcome of the course is that the students will be able to:

- Describe clearly a problem, identify its parts and analyze the individual functions.
- Apply knowledge of optimization to formulate and solve engineering problems.
- Understand the different methods of optimization and be able to suggest a technique for
  a specific problem.
- understand the effects of problem variation on the optimal solution

Topic Covered:

Introduction: Historical development, Engineering application of optimization, Formulation of design problems as mathematical programming problems, classification of optimization problems. **[10L]**

Linear Programming: Graphical method, Simplex method, Revised simplex method, Duality in linear programming, Sensitivity analysis, other algorithms for solving LP problems, Transportation Problem, Assignment Problem and other applications, Integer Programming. **[10L]**

Non Linear Programming: Unconstrained optimization techniques, Direct search methods, Descent methods, Constrained optimization, Direct and indirect methods; Optimization with calculus, Khun-Tucker conditions. **[10L]**

Introduction to Advanced Techniques of Optimization: Simulated Annealing, Genetic Algorithms, Other methods for optimization and search. **[10L]**

Books:

1. Hadley G., "Linear Programming", Narosa Publishers, 1987.
2. S. Fang et al: Linear optimizations and Extensions
3. Hadley: Linear programming, Narosa Publishing House, New Delhi, 1990.
4. K. Deb: Optimization for Engineering Design – Algorithms and Examples

MCA-491 Computer Graphics Lab

Experiments should include but not limited to :

List of Experiments:

1. Write a program to implement DDA algorithm.
2. Write a program to draw a specified figure supplied by Instructor.
3. Write a program to implement Bresenham's line algorithm.
4. What are the advantages of Bresenham's line algorithm over DDA algorithm.
5. Write a program to implement Midpoint circle gen
6. Write a program to implement Bresenham's circle generating algorithm.
7. Write a program to draw the specified figure supplied by Instructor
8. Write a program to draw the specified figure supplied by Instructor
9. Write a program to implement outline character.
10. Write a program to implement bitmap character.
11. Write a program to implement ellipse generating algorithm
12. Write a procedure to scan the interior of a specified ellipse into a solid color.
13. Write the Scan line filling algorithm.
14. Write a program to implement Line Clipping Algorithm using Cohen Sutherland Algorithm.
15. Write a program to implement Line Clipping Algorithm using Liang Barsky Algorithm.
16. Explain the Sutherland and Cohen subdivision algorithm for the line clipping.

17. Write a program to Implement Polygon Clipping Algorithm using Sutherland -Hodgman Algorithm.
18. Write a program to implement scaling on polygon.
19. Write a program to implement transferring on polygon.
20. Write a program to implement rotation on polygon.
21. Write a program to implement reflection on polygon.
22. Write a Program to implement set of Basic Transformations on Polygon i.e. Translation,Rotation and Scaling.
23. Write a program to implement set of Composite Transformations on Polygon i.e Reflection,Shear (X &Y), rotation about an arbitrary point.

MCA-492 Operating System Lab

Experiments should include but not limited to :

List of Experiments:
1. Basics of UNIX commands.
2. Shell programming
3. Implementation of CPU scheduling. a) Round Robin b) SJF c) FCFS d) Priority
4. Implement all file allocation strategies
5. Implement Semaphores
6. Implement ll File Organization Techniques a
7. Implement Bankers algorithm for Dead Lock Avoidance
8. Implement an Algorithm for Dead Lock Detection
9. Implement the all page replacement algorithms a) FIFO b) LRU c) LFU
10. Implement Shared memory and IPC
11. Implement Paging Technique f memory management.
12. Implement Threading & Synchronization Applications

MCA-493 Computer Network Lab

Experiments should include but not limited to :

List of Experiments:

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client-Server model
3. Write a code s simulating ARP/RARP protocols.
4. Write a code s simulating PING and TRACEROUTE commands
5. Create a socket for HTTP for web page upload and download.
6. Write a program to implement RPC (Remote Procedure Call)
7. Implementation of Sub-netting .
8. Applications using TCP Sockets like
   a. Echo client and echo server
   b. Chat
   c. File Transfer
9. Applications using TCP and UDP Sockets like

a. DNS
b. SNMP
c. File Transfer

10. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS

11. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.

a. Link State routing
b. Flooding
c. Distance vector

MCA-494 Compiler Design Lab

Experiments should include but not limited to :

List of Experiments:

1. Implementation of symbol table.
2. Develop a lexical analyzer to recognize a few patterns in c (ex. Identifers, constants, comments, operators etc.)
3. Implementation of lexical analyzer using lex tool.
4. Generate yacc specification for a few syntatic categories.
   a) Program to recognize a valid arithmetic expression that uses operator +, -, *, /
   b) Program to recognize a valid variable which starts with a letter followed by any number of letter or digits.
   c) Implementation of calculator using lex and yacc.
5. Implement type checking
7. Implement control flow analysis and data flow analysis.
8. Implement any one storage allocation strategies (heap, stack, static)
9. Construction of DAG
10. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using a 8086 assembler. The target assembly instructions can be simple move, add, sub, jump. Also simple addressing modes are used.
11. Implementation of simple code optimization techniques (constant folding. etc.)

THIRD YEAR FIRST SEMESTER (Semester-V)

MCA-501 Advanced Java [40L]

Objective:

Java is the most popular platform, which is used to develop several applications for the systems as well as embedded devices like mobile, laptops, tablets and many more. It is an object oriented programming language and has a simple object model, as it has derived from C and C++. It provides a virtual machine, which is accumulated with byte-code and can run on any system. With time the importance and popularity of Java is on rise as it has the magic in its remarkable abilities to innovate and morph as the technology landscape changes. It is the language of choice for developing applications for the BlackBerry Smartphone. It is important for information technology industry to develop and create multiple web-based or server based applications to enhance the industrial competency. There is huge scope for this programming language.

Outcome:

- After completing this course the student must demonstrate the knowledge and ability to:
- Able to understand the use of OOPs concepts.
- Able to solve real world problems using OOP techniques.
- Able to understand the use of abstraction.
- Able to understand the use of Packages and Interface in java.
- Able to develop and understand exception handling, multithreaded applications with synchronization.
- Able to understand the use of Collection Framework.
- Able to design GUI based applications and develop applets for web applications.

Topic Covered:

Introduction to JAVA: Class and object, basic concepts of java programming , advantages of java, byte-code & JVM. **[4L]**

Oops Concept: Inheritance, polymorphism, encapsulation, abstraction etc. **[4L]**

Class, object and data types: data types, access specifiers, operators, control statements & loops, array, creation of class, object, constructor, finalize and garbage collection. use of method overloading, this keyword, use of objects as parameter & methods returning objects, call by value & call by reference, static variables & methods, nested & inner classes, basic String handling concepts, concept of mutable and immutable string, command line arguments, **[5L]**

Java I/O: basics of I/O operations – keyboard input using Buffered Reader & Scanner classes. **[3L]**

Inheritance: Reusability – Super class & subclasses including multilevel hierarchy, process of constructor calling in inheritance, method overriding, use of super , final keyword, dynamic method dispatch, use of abstract classes & methods. **[4L]**

Java Interface and package: Use of interface, difference between interface and abstract class, creation of packages, importing packages, member access for packages. **[4L]**

Exception handling: Exception handling basics, different types of exception classes, use of try & catch with throw, throws & finally, creation of user defined exception classes. **[4L]**

Multithreading: Basics of multithreading, main thread, thread life cycle, creation of multiple threads, thread priorities, thread synchronization, deadlocks for threads, suspending & resuming threads. **[4L]**

Applet Programming: applet life cycle, difference between application & applet programming, parameter passing in applets. **[4L]**

Java AWT and Swing: concept of event model and listener, creation of GUI frames and controls like of buttons, text fields etc. **[4L]**

Books:

1. The Complete reference C++ by H. Schildt, McGrawHill
2. Learning C++: A Hands on Approach by Nagler, Jayco Publishing House
3. The C++ Programming Language by Stroustrup, Adisson Wesley
4. Object Oriented Programming in C++ by R. Lafore, SAMS
5. Java 2.0 Complete Reference by H. Schildt, McGrawHill
6. JAVA How to Program by Deitel and Deitel, Prentice Hall

MCA-502 Artificial Intelligence [40L]

Objective:

Students will try to learn:

- To create appreciation and understanding of both the achievements of AI and the theory underlying those achievements.
- To introduce the concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems
- To review the different stages of development of the AI field from human like behavior to Rational Agents.
- To impart basic proficiency in representing difficult real life problems in a state space representation so as to solve them using AI techniques like searching and game playing.

- To create an understanding of the basic issues of knowledge representation and Logic and blind and heuristic search, as well as an understanding of other topics such as minimal, resolution, etc. that play an important role in AI programs.
- To introduce advanced topics of AI such as planning, Bayes networks, natural language processing and Cognitive Computing

Outcome:

Students will able to:

- Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
- Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
- Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing
- Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.
- Formulate and solve problems with uncertain information using Bayesian approaches.
- Apply concept Natural Language processing to problems leading to understanding of cognitive computing.

Topic Covered:

Introduction: Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, Agents. **[4L]**
State Space Search: Depth First Search, Breadth First Search, DFID. **[4L]**
Heuristic Search: Best First Search, Hill Climbing, Beam Search, Tabu Search. **[6L]**
Randomized Search: Simulated Annealing, Genetic Algorithms, Ant Colony Optimization. **[3L]**
Finding Optimal Paths: Branch and Bound, A*, IDA*, Divide and Conquer approaches, Beam Stack Search. **[4L]**
Problem Decomposition: Goal Trees, AO*, Rule Based Systems, Rete Net. **[4L]**
Game Playing: Minimax Algorithm, AlphaBeta Algorithm, SSS*. **[3L]**
Planning and Constraint Satisfaction: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graphplan, Constraint Propagation. **[5L]**
Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining. **[7L]**

Books:

1. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.
2. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.

3. John Haugeland, Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985.
4. Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.
5. Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2nd edition, 2004.
6. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.
7. Elaine Rich and Kevin Knight. Artificial Intelligence, Tata McGraw Hill, 1991.
8. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
9. Eugene Charniak, Drew McDermott. Introduction to Artificial Intelligence, Addison-Wesley, 1985.
10. Patrick Henry Winston. Artificial Intelligence, Addison-Wesley, 1992.

MCA-503 Web Technology [40L]

Objective:

The objective of this paper is to develop an ability to design and implement static and dynamic website.

Outcome:

On completion of this course, a student will be familiar with client server architecture and able to develop a web application using java technologies. Students will gain the skills and project-based experience needed for entry into web application and development careers.

Topic Covered:

Introduction to the Web Technologies: Concept of WWW, Internet and WWW, HTTP Protocol: Request and Response, Web browser and Web servers. Web Security and Firewalls, Web Protocols: TCP, IP and HTTP, SMTP, POP3, FTP **[7L]**

HTML: Basics of HTML, Structure of HTML code, formatting and fonts, color, hyperlink, lists, tables, images, DOM (Programming Assignments based on above topics) **[4L]**

Style Sheets: Need for CSS, introduction to CSS, basic syntax and structure, Classes and Pseudo Classes, CSS tags for setting background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning etc. (Programming Assignments based on above topics) **[6L]**

Client Side scripting Language: (JavaScript/ VBScript etc.) and DHTML. **[3L]**

Introduction to PHP: Configuration and Installation of PHP, basic syntax of PHP, Expressions, Statements, Arrays, Functions, string, Regular Expressions, Date and Time Functions (Programming Assignments based on above topics) **[3L]**

PHP and MySQL: File Handling- Creating a File, Reading from Files, Copying Files, Moving File, Deleting File, Updating File, Uploading Files, Form Designing using HTML 5, Validation's using PHP Connection to server, creating database, selecting a database, listing database, listing table names, creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, Master-Detail relationships using Joins. Session Management- Using Cookies in PHP, HTTP Authentication, Using Sessions. (Programming Assignments based on above topics) **[12L]**

Web services: Design and modeling of web services, Technologies for implementing web services **[5L]**

Books:

1. Web Technologies, Black Book, Dreamtech Press
2. Learning PHP, MySQL, JavaScript, CSS and HTML 5, Robin Nixon, O'Reilly publication
3. Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
4. Professional PHP Programming, Jesus Caspagnetto, Etal. Wrox Publication.
5. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson
6. Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

MCA-504 Elective-I

(A) Image Processing [40L]

Course Objectives

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.

Course Outcomes

- Review the fundamental concepts of a digital image processing system.
- Analyze images in the frequency domain using various transforms.
- Evaluate the techniques for image enhancement and image restoration.
- Categorize various compression techniques.
- Interpret Image compression standards.
- Interpret image segmentation and representation techniques.

Topic Covered:

Fundamentals of Digital Image Processing, Image representation, Basic Image transforms, image file format **[4L]**
Image Enhancement: Contrast stretching, Histogram Equalization, Binarization **[4L]**
Filtering in Spatial domain: Mean filter, Order Statistics filters. **[2L]**
Filtering in Frequency domain : Butterworth filter, Gaussian filter. **[2L]**
Image Restoration : Image degradation models, Weiner filter. **[2L]**
Image textures: Run Length Coding, Gray-level co-occurrence matrix **[2L]**
Image Segmentation:
Edge detection: Gradient operators, Compass operator, Laplacian operators. LoG operator. **[4L]**
Region Segmentation : Region growing, region splitting and merging. **[2L]**
Shape detection: Least Mean Square error line fitting, Eigenvector line fitting, Straight line Hough Transform, Generalized Hough Transform. **[4L]**
Morphological Operators:
Dilation, Erosion, Opening , Closing, Hit-and-Miss transforms, Applications. **[4L]**
Image Compression. **[4L]**
Image Understanding:
Feature extraction techniques, Statistical Decision making techniques, Nearest Neighbour Clustering, Maxi-min Clustering, Discriminant functions, Artificial Neural Networks. **[6L]**

Books:
1. Digital Image Processing: R.C. Gonzalez and R. E. Woods, Pearson Education.
2. Pattern Recognition and Image Analysis: E. Gose, R. Johnsonbaugh, Steve Jost, Prentice Hall India.
3. B. Chanda and D. Dutta Majumder, “Digital Image Processing and Analysis”, PHI.
4. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI.
5. M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis and Machine Vision”, Thomson Learning, 1999.
6. Malay K. Pakhira, Digital Image Processing and pattern recognition, PHI, 2011

(B) Distributed computing [40L]

Outcome:

- The student will explain various architectures used to design distributed systems, such as client-server and peer-to-peer.
- The student will build distributed systems using various interprocess communication techniques, such as remote method invocation, remote events, and tuple spaces.
- The student will build distributed systems using various techniques for tolerating partial failures, such as leasing and replication.
- The student will build distributed systems using various interprocess coordination techniques, such as distributed mutual exclusion, distributed monitors, and tuple spaces.
- The student will explain various distributed algorithms, such as logical clocks and leader election.

Topic Covered:

Introduction to distributed environment: **[4L]**
Goals, hardware & software concepts, P2P, Cluster, Grid, Cloud, the client-server model, Strengths and weakness of distributed computing, forms of coputing
Communication: **[4L]**
Layered protocols, RPC, remote object invocation, message-oriented communication
Distributed computing paradigms: **[5L]**
Message passing, client server, P2P, remote procedure call model, distributed objects, object space, collaborative application (groupware)
Socket: **[2L]**
Socket metaphor, datagram socket API, stream mode socket API, sockets with non blocking I/O, secure socket API
Java RMI: **6L]**
Client side, Server Side, object registry, Remote Interface, Server side software, client side software, RMI vs Socket
Advanced RMI: **[3L]**
Client callback, stub downloading, RMI security manager
Group Communication: **[2L]**
Unicasting, multicasting, connection oriented & connectionless, reliable and unreliable multicast, Java basic multicast API
Internet Applications: **[5L]**
HTML, XML, HTTP, Applets, Servlets, Web services, SOAP
Mobile Agents: **[2L]**
Basic architecture, advantages, mobile agent framework systems, design, implementation using Java RMI
Distributed coordination-based systems JINI: **[5L]**
Runtime environment, architecture, discovery protocol, join protocol, lookup service, distributed event, distributed leasing, transactions, surrogate architecture
New paradigms of distributed computing environment **[2L]**

Books:
1. Distributed Computing: Principles and Applications, M. L. Liu, Pearson/Addison-Wesley.
2. A Programmer's Guide to Jini Technology, Jan Newmarch, Apress.
3. A. Taunenbaum, Distributed Systems: Principles and Paradigms, PHI
4. G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, Pearson Education
5. Core Jini, W. Kieth Edwards, Apress.

(C) Mobile Computing [40L]

Learning Outcomes

Upon successful completion of this course, you will be able to

- explain the principles and theories of mobile computing technologies.
- describe infrastructures and technologies of mobile computing technologies.

- list applications in different domains that mobile computing offers to the public, employees, and businesses.
- describe the possible future of mobile computing technologies and applications.

Topic Covered:

Introduction: Introduction to wireless networks and mobile computing – Characteristics, Issues and challenges **[5L]**

Wireless Transmission: Fundamentals of wireless transmission - Medium Access Control Protocols, Different types of multiple access techniques and their characteristics **[5L]**

Cellular Communication: Cellular concept, Overview of different Generations **[5L]**

Mobile: Mobile IP, Mobile transport layer - Mechanisms for improving TCP performances on wireless links, , Overview of Security in mobile environments **[10L]**

Wireless: Overview of Wireless LAN IEEE 802.11 series, Overview of Bluetooth, Overview of Wireless Sensor Networks **[5L]**

Wireless application Environments: WAP, WML, Push Architecture, Push/Pull Services Mobile Adhoc Networks – Characteristics, Routing protocols **[10L]**

Books:

1. Mobile Computing, Raj Kamal, Oxford
2. Hansmann, Merk, Nicklous, Stober, "Principles of Mobile Computing", Springer, second edition, 2003.
3. Mobile Communications, Jochen Schiller, Pearson Education
4. Stojmenovic and Cacute, "Handbook of Wireless Networks and Mobile Computing", Wiley, 2002, ISBN 0471419028.

(D) Pattern Recognition [40L]

Outcome

At the end of this course, students will be able to:

- Explain and compare a variety of pattern classification, structural pattern recognition, and pattern classifier combination techniques.
- Summarize, analyze, and relate research in the pattern recognition area verbally and in writing.
- Apply performance evaluation methods for pattern recognition, and critique comparisons of techniques made in the research literature.
- Apply pattern recognition techniques to real-world problems such as document analysis and recognition.
- Implement simple pattern classifiers, classifier combinations, and structural pattern recognizers.

Topic Covered:

Basic concepts of Pattern Recognition **[2L]**
Pattern Preprocessing and Feature Selection **[3L]**
Decision Functions **[2L]**
Bayesian decision theory **[2L]**
Parametric Estimation: Maximum likelihood estimation and Bayesian estimation **[5L]**
Non- parametric Estimation: Parzen windows, Nearest Neighbor estimation **[4L]**
Pattern Classification:
Linear classifier: Perceptron, SVM [**3L]**
Non-linear classifiers: MLP, Non-linear SVM [**5L]**
Unsupervised learning and Clustering: Partitioning method, Density-based method, MST-basedmethod, Self organizing map, Hierarchical Clustering, Cluster validity **[10L]**
Syntactic Pattern Recognition (Basic concepts) **[2L]**
Some real-life applications **[2L]**

Books:

1. Pattern Recognition Principles, Tou and Gonzalez, Addison-Wesley
2. Pattern Classification, Duda, Hart and Stork, Second Edition, Wiley
3. Pattern Recognition and Machine Learning, Christopher Bishop, Springer
4. Introduction to Statistical Pattern Recognition, Fukunaga, Second Edition, Academic Press

(E)Machine Learning [40L]

Outcome
On completion of the course students will be expected to:

- Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
- Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
- Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
- Be able to design and implement various machine learning algorithms in a range of real-world applications.

Topic Covered:

Introduction: **[3L]**
Machine learning applications, concepts learning
Introduction to Bayesian learning theory: **[5L]**
regression, feature selection, supervised learning, class conditional probability distributions, Examples of classifiers Bayes optimal classifier and error, learning classification approaches, handling continuous attributes.
Decision tree learning algorithms: **[6L]**
Inference model, general domains, symbolic decision trees, consistency, learning trees from training examples, entropy, mutual information, ID3 algorithm criterion, C4.5 algorithm, handling continuous and missing attributes, confidence, overfitting, pruning, learning with

incomplete data

Artificial Neural Network: [**4L]**

Single layer neural network, linear reparability, general gradient descent, perceptron learning algorithm, multi-Layer perceptron: two-layers universal approximators, backpropagation learning, important parameters, Margin of a classifier, dual perceptron algorithm, learning nonlinear
hypotheses with perceptron.

Instance-based Learning: **[2L]**

Nearest neighbor classification, k-nearest neighbor, nearest neighbor error probability,

Machine learning concepts and limitations: **[10L]**

Learning theory, formal model of the learnable, sample complexity, learning in zero-bayes and realizable case, VC-dimension, fundamental algorithm independent concepts, hypothesis class, target class, inductive bias, occam's razor, empirical risk, limitations of inference machines, approximation and estimation errors, Tradeoff.

Support Vector Machine (SVM): **[3L]**

Kernel functions, implicit non-linear feature space, theory, zero-Bayes, realizable infinite hypothesis class, finite covering, margin-based bounds on risk, maximal margin classifier.

Machine learning assessment and Improvement: **[3L]**

Statistical model selection, structural risk minimization, bootstrapping, bagging, boosting.

Unsupervised learning: **[2L]**

introduction, K- means clustering, Hierarchical clustering

Semi-supervised learning: **[2L]**

introduction, self-training, co-training

Books:

1. T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.
2. E. Alpaydin, Introduction to Machine Learning, Prentice Hall of India, 2006.
3. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
4. R. O. Duda, P. E. Hart, and D.G. Stork, Pattern Classification, John Wiley and Sons, 2001.
5. Vladimir N. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
6. Shawe-Taylor J. and Cristianini N., Cambridge, Introduction to Support Vector Machines, University Press, 2000.

(F)Soft Computing [40L] 1.

Objective:

The primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of soft computing.

Upon successful completion of the course, students will have an understanding of the basic areas of Soft Computing including Artificial Neural Networks, Fuzzy Logic and Genetic Algorithms.

Provide the mathematical background for carrying out the optimization associated with neural network learning.

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.

Outcome:

The student will be able to:

- Describe human intelligence and AI
- Explain how intelligent system works.
- Apply basics of Fuzzy logic and neural networks.
- Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience
- Relate with neural networks that can learn from available examples and generalize to form appropriate rules for inference systems
- Describe with genetic algorithms and other random search procedures useful while seeking global optimum in self-learning situations
- Develop some familiarity with current research problems and research methods in Soft Computing Techniques.

Topic Covered:

Introduction to Soft Computing, Components of Soft Computing, Importance of Soft Computing, Applications. [**2L**]
2. Fuzzy Set Theory - Definition, Different types of fuzzy set membership functions. Fuzzy set theoretic operations, Fuzzy rules and fuzzy reasoning, Fuzzy inference systems. [**8L**]
3. Rough set theory. [**2L**]
4. Probabilistic Reasoning. [**4L**]
5. Genetic Algorithms, Simulated Annealing, applications. [**6L**]
6. Neural Networks- Artificial neural networks models, Supervised Learning, Unsupervised Learning, Applications. [**10L**]
7. Hybrid Systems and applications [**8L**]

Books:

1. Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence - Jang, Sun and Mizutani, Printice Hall.
2. Soft Computing : Integrating Evolutionary, Neural, and Fuzzy Systems, by Tettamanzi, Andrea, Tomassini, and Marco. (2001), Springer.

(G)Embedded System [40L]

Objective:

- To have knowledge about the basic working of a microcontroller system and its programming in assembly language.
- To provide experience to integrate hardware and software for microcontroller applications systems.

Outcome:

To acquire knowledge about microcontrollers embedded processors and their applications.

- Foster ability to understand the internal architecture and interfacing of different
- peripheral devices with Microcontrollers.
- Foster ability to write the programs for microcontroller.
- Foster ability to understand the role of embedded systems in industry.
- Foster ability to understand the design concept of embedded systems.

Topic Covered:

Introduction- Embedded system overview, embedded hardware units, embedded software in a system, embedded system on chip (SOC), design process, classification of embedded systems **[7L]**

Embedded computing platform - CPU Bus, memory devices, component interfacing, networks for embedded systems, communication interfacings: RS232/UART, RS422/RS485, IEEE 488 bus. **[5L]**

Survey of software architecture- Round robin, round robin with interrupts, function queue scheduling architecture, selecting an architecture saving memory space **[5L]**

Embedded software development tools- Host and target machines, linkers, locations for embedded software, getting embedded software into target system, debugging technique **[5L]**

RTOS concepts - Architecture of the kernel, interrupt service routines, semaphores, message queues, pipes **[4L]**

Instruction sets- Introduction, preliminaries, ARM processor, SHARC processor. **[4L]**

System design techniques - Design methodologies, requirement analysis, specifications, system analysis and architecture design **[5L]**

Design examples- Telephone PBX, ink jet printer, water tank monitoring system, GPRS, Personal Digital Assistants, Set Top boxes. **[5L]**

Books:

1. Computers as a component: principles of embedded computing system design- wayne wolf
2. An embedded software premier: David E. Simon
3. Embedded / real time systems-KVKK Prasad, Dreamtech press, 2005
4. Programming for embedded system by Dr. Prasas, Vikas Gupta, Das & Verma, Pub, WILEY Dreamtech india Pvt.
5. Embadded System Design. by Frank Vashid & Tony Givergis, Pub, WILEY.
6. MFC Programming. by Herbert Schildt, Pub. TataMcGraw Hill.
7. Programming Embedded Systems by Michael Barr, Pub. O'REILLY

(H) Cloud Computing [40L]

Objective:

- To learn how to use Cloud Services.
- To implement Virtualization
- To implement Task Scheduling algorithms.
- Apply Map-Reduce concept to applications.
- To build Private Cloud.
- Broadly educate to know the impact of engineering on legal and societal issues involved.

Outcome:

The primary learning outcomes of this course are five-fold. Students will be able to:

- Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came
- about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
- Apply fundamental concepts in cloud infrastructures to understand the tradeoffs in power, efficiency and cost, and then study how to leverage and manage single and multiple datacenters to build and deploy cloud applications that are resilient, elastic and cost-efficient.
- Discuss system, network and storage virtualization and outline their role in enabling the cloud computing system model.
- Illustrate the fundamental concepts of cloud storage and demonstrate their use in storage systems such as Amazon S3 and HDFS.
- Analyze various cloud programming models and apply them to solve problems on the cloud.

Topic Covered:

Introduction: Essentials, Benefits and need for Cloud Computing - Business and IT Perspective - Cloud and Virtualization - Cloud Services Requirements - Cloud and Dynamic Infrastructure - Cloud Computing Characteristics Cloud Adoption. **[3L]**

Cloud Models: Cloud Characteristics - Measured Service - Cloud Models - Security in a Public Cloud Public versus Private Clouds - Cloud Infrastructure Self Service **[3L]**

Cloud as a Service: Gamut of Cloud Solutions - Principal Technologies - Cloud Strategy Cloud Design and Implementation using SOA - Conceptual Cloud Model - Cloud Service Defined **[4L]**

Cloud Solutions: Cloud Ecosystem - Cloud Business Process Management - Cloud Service Management - Cloud Stack - Computing on Demand (CoD) – Cloud sourcing. **[3L]**

Cloud Offerings: Information Storage, Retrieval, Archive and Protection - Cloud Analytics Testing under Cloud - Information Security - Virtual Desktop Infrastructure - Storage Cloud. **[3L]**

Cloud Management: Resiliency – Provisioning - Asset Management - Cloud Governance - High Availability and Disaster Recovery - Charging Models, Usage Reporting, Billing and Metering. **[5L]**

Cloud Virtualization Technology: Virtualization Defined - Virtualization Benefits - Server Virtualization - Virtualization for x86 Architecture - Hypervisor Management Software - Logical Partitioning (LPAR) - VIO Server - Virtual Infrastructure Requirements. **[5L]**

Cloud Virtualization: Storage virtualization - Storage Area Networks - Network-Attached storage - Cloud Server Virtualization - Virtualized Data Center. **[5L]**

Cloud and SOA: SOA Journey to Infrastructure - SOA and Cloud - SOA Defined - SOA and IaaS - SOA-based Cloud Infrastructure Steps - SOA Business and IT Services. **[5L]**

Cloud Infrastructure Benchmarking: OLTP Benchmark - Business Intelligence Benchmark - e-Business Benchmark - ISV Benchmarks - Cloud Performance Data Collection and Performance Monitoring Commands - Benchmark Tools. **[4L]**

Book:

1. Cloud Computing – Insight into New Era Infrastructure, Dr. Kumar Saurabh, Wiley India.
2. Cloud Computing Explained, John Rhoton, Recursive Press

3. Cloud Computing Bible, Barry Sosinsky, Wiley
4. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg, Wiley
5. Cloud Computing for Dummies, Judith Hurwiz, Wiley Publishing.
6. The Cloud at your service, Rosenberg and Matheos, Manning Publications

MCA-505 Elective-II

(A) Multimedia [40L]

Objective:
Students will try to learn:

- To learn and understand technical aspect of Multimedia Systems.
- To understand the standards available for different audio, video and text applications.
- To Design and develop various Multimedia Systems applicable in real time.
- To learn various multimedia authoring systems.
- To understand various networking aspects used for multimedia applications.
- To develop multimedia application and analyze the performance of the same

Outcome:
Students will able to:

- Developed understanding of technical aspect of Multimedia Systems.
- Understand various file formats for audio, video and text media.
- Develop various Multimedia Systems applicable in real time.
- Design interactive multimedia software.
- Apply various networking protocols for multimedia applications.
- To evaluate multimedia application for its optimum performance.

Topic Covered:

Introduction
Multimedia and its Application, Different Media, Hypertext and Hypermedia, Issues in Multimedia System, Component of a Multimedia System **[2L]**

Overview of Text and Graphics:
Types of Text Data (Plain/Formatted/Hypertext), Unicode Scheme, Concept of Font, File Formats (txt, doc, rtf, ps, pdf etc.), Vector and Raster Graphics **[2L]**

Image:
Image Digitization, Digital Image, Binary/GrayScale/ Colour Image, Colour Models, File Formats, Overview of Contrast Intensification, noise removal, edge detection and segmentation **[5 L]**

Image Descriptors (Shape, Texture and Colour Features) **[3L]**

Loss-less and Lossy Image Compression including JPEG **[3L]**
An overview of Content Based Image Retrieval System **[3L]**

Audio:

Audio Digitization (Sampling and Quantization, Representation based on PCM/DPCM/DM/ADM), File Formats **[2L]**
Time Domain Descriptors (ZCR, STE etc.), Frequency Domain Descriptors (Spectral Centroid, Spectral Flux, Spectral Roll Off etc.), and Perception based Descriptors (Mel Scale, MFCC) **[3 L]**
Psycho Acoustics and Audio Compression **[2L]**
An Overview of Audio Classification/Retrieval System **[2L]**

Video:

Structure of Video Data, File Formats **[1L]**
Video Compression **[2L]**
Motion Estimation **[1L]**
Structural Segmentation of Video Data **[3L]**
Overview of Video Summarization, Browsing and Retrieval System **[2L]**

Animation:

Keyframes & tweening, cel & path animation, principles and techniques of animation, Web animation, 3D animation principles, camera, special effects, transformations and editing, rendering algorithms, features of animation software, file formats. **[4L]**

Books:

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods
2. Digital Image Processing and Analysis by B. Chanda and D. Dutta Majumder
3. Principles of Multimedia by Ranjan Parekh
4. Multimedia –A Practical Approach by Sanhker, Jaico.
5. Multimedia Systems by Buford J. K., Pearson Education.
6. Multimedia and Imaging Databases by S. Khoshafian, A. Brad Baker, Morgan Kaufmann.
7. Multimedia Systems Design, Prabhat k. Andleigh & Kiran Thakkar, Prentice Hall PTR.
8. Digital Multimedia by Nigel Chapman & Jenny Chapman, John-Wiley.

(B) Data Mining [40L]

Objective:

Students undergoing this course are expected to:

- Differentiate OnLine Transaction Processing and OnLine Analytical processing
- Learn Multidimensional schemas suitable for data warehousing
- Understand various data mining functionalities
- Inculcate knowledge on data mining query languages.
- Know in detail about data mining algorithms

Outcome:

After undergoing the course, Students will be able to understand

- Design a data mart or data warehouse for any organization
- Develop skills to write queries using DMQL
- Extract knowledge using data mining techniques
- Adapt to new data mining tools.
- Explore recent trends in data mining such as web mining, spatial-temporal mining

Topic Covered

Introduction: concepts of data mining,	[**2L**]
Differences of conventional Database with data warehouse	**[2L]**
Concepts of Data Cubes and OLAP Data Processing	**[3L]**
Types of Data Warehouses and Schemas	**[2L]**
Development Methodologies	**[4L]**
Management of Data Warehouses	**[2L]**
Data Mining Algorithms in general with scalability issues	**[2L]**
Evaluation of data mining results	**[1L]**
Data Pre-processing Techniques	**[5L]**
Application of Association Rule Mining in data mining	**[5L]**
Application of Clustering Algorithms in data mining	**[4L]**
Application of Classification Algorithms in data mining	**[4L]**
Text mining, Web mining and other Applications	**[3L]**
Recent Trends	**[1L]**

Books:

1. J. Han & M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2nd Ed.
2. Data warehousing: OLAP & data mining, S. Nagabhushan, New age publications.
3. Introduction to data mining by Tan, Steinbach, Kumar, Pearson Education
4. Data mining: A tutorial based primer by Roiger, Geatz,, Pearson Education

(C) Parallel Computing [40L]

Objective:

Students will demonstrate an understanding of concepts, algorithms, and design principles underlying parallel computing, develop algorithm design and implementation skills, and gain practical experience in programming large scale parallel machines

Outcome:

The course outcomes are skills and abilities students should have acquired by the end of the course.

- Students demonstrate they can define and apply parallel computing to a variety of applications in Mathematics and Engineering.
- Students will have an ability to assess a problem presented to them, design a solution, and test their implementation.
- Students will be presented with problems and will have to design and implement solutions for those problems
- Students will have an ability to discuss large scale machine design as well as applications and algorithms on those machines.

Topic Covered

Introduction to High Performance Computing: Milestones and applications. **[4L]**
High-Performance Computing architectures: Overview of the major classes of HPC architectures and their evolution. **[6L]**
Parallel programming models and performance analysis: Parameterisation, modeling, performance analysis, Amdahl's law, efficiency, and benchmarking of systems. **[4L]**
Programming parallel computers: Overview of parallel programming, parallel languages,

parallelizingcompilers, message passing and data parallel programming models, introduction to MPI and OpenMP. **[10L]**

Multi-Thread Models with primary sources of overhead, memory architecture and memory access times and associated sources of overhead; Multi-Process Execution Model. Restructuring for Parallel Performance - Loop Transformations; Data Transformations; Dependence Analysis; Compiler Strategies. **[8L]**

Parallel Algorithms - Cyclic Reduction; Iterative Algorithms (Jacobi, Gauss-Seidel and Red-BlackOrderings); Divide-and-Conquer Algorithms, Adaptive Quadrature. **[8L]**

Books:

1. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey
2. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series,
3. Parallel Programming in C with MPI and OpenMP by M.J. Quinn, McGraw-Hill

(D) Cryptography and Steganography [40L]

Course objective:

- To learn about the threats of network security.
- To understand what causes these threats by studying how vulnerabilities arise in the development and uses of computer system.
- To understand the architecture of network security.
- To narrate and evaluate the design principles of conventional encryption and decryption techniques.
- To analyze the concepts of public key encryption and public key algorithm.
- To narrate and evaluate the design principles of conventional data hiding and watermarking schemes

Course outcomes:

- Ability to analyze and determine for any organization the security requirements and appropriate solutions.
- Ability to protect system from different types of threats, malicious software's vulnerabilities and attacks.
- Ability to describe symmetric and public key encryption algorithms like DES, AES, RSA etc.
- Ability to identify ethical, professional responsibilities, risks and liabilities in computer and network environment, and best practices to write security policy.
- Ability to narrate the Authentication of digital certificates.
- Ability to differentiate MAC and hashing techniques needed for authentication.
- Ability to protect information from different types of threats, malicious software's vulnerabilities and attacks through Data hiding and Watermarkin scheme.

Topic Covered:

Introduction to security attacks, services and mechanism, introduction to cryptography.

Conventional Encryption: Conventional encryption model, classical encryption

techniques- substitution ciphers and transposition ciphers, cryptanalysis, stereography, **[6L]**

Stream and block ciphers.
Modern Block Ciphers: Block ciphers principals, Shannon's theory of confusion and diffusion, fiestal structure, data encryption standard(DES), strength of DES, differential and linear crypt analysis of DES, block cipher modes of operations, triple DES, IDEA encryption and decryption, strength of IDEA, confidentiality using conventional encryption, traffic confidentiality, key distribution, random number generation **[6L]**

Introduction to Information Hiding: Technical Steganography, Linguistic Steganography, Copy Right Enforcement, Wisdom from Cryptography Principles of Steganography: Framework for Secret Communication, Security of Steganography System, Information Hiding in Noisy Data , Adaptive versus non-Adaptive Algorithms, Active and Malicious Attackers, Information hiding in Written Text. **[10L]**

A Survey of Steganographic Techniques: Substitution systems and Bit Plane Tools, Transform Domain Techniques: - Spread Spectrum and Information hiding, Statistical Steganography, Distortion Techniques, Cover Generation Techniques.
Steganalysis: Looking for Signatures: - Extracting hidden Information, Disabling Hidden Information. **[10L]**
Watermarking and Copyright Protection: Basic Watermarking, Watermarking Applications,
Requirements and Algorithmic Design Issues, Evaluation and Benchmarking of Watermarking system. **[8L]**

Books:

1. William Stallings, "Cryptography and Network Security: Principals and Practice", Prentice Hall, New Jersy.
2. Johannes A. Buchmann, "Introduction to Cryptography", Springer-Verlag.
3. Bruce Schiener, "Applied Cryptography".
4. Katzendbisser, Petitcolas, " Information Hiding Techniques for Steganography and Digital Watermarking", Artech House.
5. Peter Wayner, "Disappearing Cryptography: Information Hiding, Steganography and Watermarking 2/e", Elsevier
6. Bolle, Connell et. al., "Guide to Biometrics", Springer

(E) Bio Informatics [40L]

Objective:
Students undergoing this course are expected to:

- Understand the basic concepts and techniques of Bioinformatics.
- Develop an awareness of the computational problems that arise in the modeling and analysis of living systems.
- Understand basic abstractions and computational approaches used to formulate and address these problems.

Outcome:
After undergoing the course, Students will be able to understand

- Sequencing Alignment and Dynamic Programming

- Sequence Databases

Topic Covered

Introduction to molecular biology, The Central Dogma of Molecular Biology, Physical mapping **[4L]**

Protein sequence data bank. NBFR-PIR, SWISSPROT, GenBank, EMBL nucleotide sequence data bank, Protein Data Bank (PDB) etc. **[3L]**

Motif finding in DNA and proteins **[4L]**

Sequence alignment for DNA and protein sequences, Concepts: homology, sequence similarity and sequence alignment; dynamic programming algorithms, Pairwise alignment, Global and local alignment using dynamic programming, Heuristic alignment methods: BLAST/FASTA and the statistics of local alignments, Multiple sequence alignment: Definition, scoring, techniques, Aligners for proteins sequences, Spliced alignment **[5L]**

Gene ontology, Annotation and Metadata **[5L]**

Secondary and Tertiary Structure predictions; Chao-Fasman algorithms; The basic HMM algorithms: forward, backward, Viterbi, Baum-Welch; Neural Networking **[5L]**

Phylogenetic analysis, Neighbor joining, parsimony, and maximum likelihood methods, Gene expression analysis and clustering methods **[5L]**

Comparative genomics: gene regulation, gene finding, genome rearrangements **[4L]**

Protein folding, protein-protein interactions, Molecular Modeling & Dynamics, Drug Designing. **[5L]**

Books:

[1] M. Lesk, "Introduction to Bio Informatics," Oxford University Press

[2] Hooman Rashidi, Lukas K. Buehler, "Bioinformatics Basics: Applications in Biological Science and Medicine," CRC Press/Taylor & Francis Group, 2nd edition, May 2005

[3] Jeffrey Augen, "Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and
Information-Based Medicine," Addison-Wesley

[4] Stephen A. Krawetz, David D. Womble, "Introduction to Bioinformatics: A Theoretical and Practical
Approach," Humana Press

[5] Bryan Bergeron, "Bioinformatics Computing," Prentice Hall PTR

[6] Malcolm Campbell, Laurie J. Heyer, "Discovering Genomics, Proteomics, and Bioinformatics,"
Benjamin/Cummings

[7] Teresa K. Attwood, David Parry-Smith, "Introduction to Bioinformatics," Pearson Education

[8] Gusfield, Dan, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational
Biology," Cambridge, UK: Cambridge University Press, 1997.

[9] Waterman, Michael, "Introduction to Computational Biology: Maps, Sequences, and Genomes,"
Boca Raton, FL: CRC Press, 1995.

[10] Durbin, Richard, Graeme Mitchison, S. Eddy, A. Krogh, and G. Mitchison, "Biological Sequence
Analysis: Probabilistic Models of Proteins and Nucleic Acids," Cambridge, UK: Cambridge University Press, 1997.

[11] Jones, Neil, and Pavel Pevzner, "An Introduction to Bioinformatics Algorithms," Cambridge,
MA: MIT Press, 2004.

(F) Natural Language Processing [40L]

Objective:

To understand natural language processing and to learn how to apply basic algorithms in this field. To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

Outcome:

The students will get acquainted with natural language processing and learn how to apply basic algorithms in this field. They will understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. They will also grasp basics of knowledge representation, inference, and relations to the artificial intelligence

Topic Covered

Speech & Natural Language Processing:	**[1L]**
Brief Review of Regular Expressions and Automata; Finite State Transducers;	**[3L]**
Word level Morphology and Computational Phonology;	**[3L]**
Basic Text to Speech;	**[3L]**
Introduction to HMMs and Speech Recognition.	**[3L]**
Indian language case studies; Part of Speech Tagging;	**[3L]**
Parsing with CFGs; Probabilistic Parsing.	**[2L]**
Representation of Meaning;	**[3L]**
Semantic Analysis; Lexical Semantics; Word Sense;	**[3L]**
Disambiguation; Discourse understanding;	**[3L]**
Natural Language Generation.	**[3L]**

Book:

1. Natural Language Processing And Information Retrieval, TANVEER SIDDIQUI, U. S TIWARY, Oxford University Press
2. NATURAL LANGUAGE UNDERSTANDING, J Allen, Pearson India
3. Multilingual Natural Language Processing Applications from Theory to Practice, Bikel, Pearson India
4. NATURAL LANGUAGE PROCESSING, Dipti Mishra Sharma, MACMILLAN INDIA LTD

(G) Computational Geometry [40L]

Objective:

The objectives of this course are as follows:

- Introduce rigorous algorithmic analysis for problems in Computational Geometry.
- Discuss applications of Computational Geometry to graphical rendering.
- Introduce the notions of Voronoi diagrams and Delaunay Triangulations.

- Develop expected case analyses for linear programming problems in small dimensions.

Outcome:

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

- Analyze randomized algorithms for small domain problems.
- Use line-point duality to develop efficient algorithms.
- Apply geometric techniques to real-world problems in graphics.
- Solve linear programs geometrically

Topic Covered

Geometric Objects – Points, Lines, Planes, Polygons, 3D Objects – Geometric Algorithms – Degeneracies and Robustness – Application Domains [**3L]**

Convex Hull in 2D – Incremental Algorithm [**2L]**

Line Segment Intersection Algorithms – Doubly Connected Edge List – Map Overlays – Boolean operations [**6L]**

Polygon Triangulation – Partitioning Polygons into Monotone Pieces – Triangulation of Monotone Polygons – Art Gallery Problem [**6L]**

Half Plane Intersections – Use of Linear Programming Techniques – Manufacturing with Moulds [**6L]**

Orthogonal Range Searching – Kd Trees – Range Trees – Higher Dimensional Range Trees Database Searching – Point Location [**6L]**

Voronoi Diagrams – VD of Line Segments – Farthest Point VDs – Post Office Problem 6L

Convex Hulls in 3-space [**3L]**

Robot Motion Planning – Work Space and Configuration Space – Translational Motion Planning [**2L]**

Books:

1. Computational Geometry – Algorithms and Applications by Berg, Cheong, Kreveld and Overmars 3e, Springer
2. Computational Geometry – An Introduction by Preparata and Shamos, Springer
3. Computational Geometry in C – Joseph O'Rourke, 2e, Cambridge Univ Press

(H) Recent Computing Architecture Trends [40L]

Distributed computing architectures: Introduction, Client-server architecture, Peer-to-peer systems, Applications. **[10L]**

Parallel and scalable architectures: Multiprocessors and multicomputer architectures, Multi-vector and SIMD computers, Scalable, multithreaded and data flow architectures. **[10L]**

Grid computing architectures: Introduction, Benefits, terms and concepts, grid user roles, grid architecture considerations, standards for grid environments, applications. **[10L]**

Cloud computing architectures: Introduction, Layers of cloud architecture, understanding cloud ecosystem, cloud architecture components, applications. **[10L]**

Books:

1. Cloud Computing for Dummies by Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper (Wiley India Edition)
2. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
3. *Distributed Computing*. Principles, Algorithms, and. Systems. Ajay D. Kshemkalyani. Mukesh Singhal. Cambridge University press
4. Principles of Grid Computing, Krishna, Paperback

MCA-591 Java Lab

Experiments should include but not limited to :

List of Assignments:

1. Assignments on class, constructor, final, super and static keywords.
2. Assignments on function overloading and overriding.
3. Assignments on pointers, arrays, interface, abstract class and package.
4. Assignments on exception handling and multithreading.
5. Assignments on String handling and file handling.
6. Assignments on Applet and AWT.

MCA-592 AI Lab

Experiments should include but not limited to :

List of Assignments:

1. Study of PROLOG.
2. Write the following programs using PROLOG:
3. Write a program to solve 8-queens problem.
4. Solve any problem using depth first search.
5. Solve any problem using best first search.
6. Solve 8- puzzle problem using best first search.
7. Solve Robot (traversal) problem using means End Analysis.
8. Solve Traveling Salesman problem.

MCA-593 Web Technology Lab

Experiments should include but not limited to :

List of Assignments:

Problems and assignments based on Web Technology i.e. PHP (HyperText Preprocessor) language.
Form creation, Simple CSS creation. Data insertion, updation, deletion through form into a MySQL database. Data selection from MySQL Database and view on a web page. Data Authentication using session, cookie.
Some Experiments:

1. Create a web page to validate an arithmetic captcha. Use session to validate a captcha.
   Ca : 5+3 =
2. Create a web page to edit the following information and store the updated information in database. Create the database.
   Name: Text Box
   Subject: Check Box
3. Design a web page to implement user registration process. User submits username, password, phone no., email. Also, check the availability of the username. After successful registration, print a success message.
4. Create a student database using MYSQL with name, roll, marks of three subjects (MATH, SCIENCE, ENGLISH).Create a page in PHP to access the table against roll and to generate the average marks of the student and display the average marks.
5. Design a login page for an administrator. If authentication is correct allow him to visit any page (dummy) else send him to login page.

MCA-594 (A) Term Paper-II

Seminar topic will be assigned to individual student by the Head of the department at the beginning of the semester.

MCA-594 (B) Grand Viva

Each student will appear for Viva-Voce before the examiners. Viva-Voce will be conducted based on all Theoretical subjects of this curriculum.

THIRD YEAR SECOND SEMESTER (Semester-VI)

MCA-691 Project Work / Dissertation

Student will be assigned for six months project in any IT industry / Educational Institute.

MCA-692 Seminar Presentation

Student will deliver seminar presentation based on the Project Work / Dissertation.

MCA-693 Group Discussion

Students have to appear for group discussion on current topics provided by the Examiners.