SCHEME & SYLLABUS FOR PROGRAM B.TECH COMPUTER ENGINEERING

Bachelor of Technology (Computer Engg)
Scheme of Courses/Examination
(1 st SEMESTER)

(Common to all Branches)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P/D	Total	Theory	Sessional	P/VV	Total	
1.	HUT-109	Engineering Economics **OR**	3	1	-	4	50	50	-	100	3
2.	MET-103	Manufacturing Processes	3	1	-	4	50	50	-	100	3
3.	HUT-107	Commun. Skills in English **OR**	2	2	-	4	50	50	-	100	3
4.	ECT-103	Basic Electronics Eng. and	2	2/2	-	3	50	50	-	100	3
5.	ECT-105	Basic Electronics Eng. (Pr)	-	-	2/2	1		60	40	100	3
6.	MAT-103	Mathematics-I	3	2	-	5	50	50	-	100	3
7.	PHT-105	Physics-I	3	1	-	4	50	50	-	100	3
8.	CHT-105	Chemistry – I	3	1	-	4	50	50	-	100	3
9.	ELT-105	Basic Electrical Engineering **OR**	2	2/2	-	3	60	40	-	100	3
10.	COT-101	Computer Engineering *	2	2/2	-	3	50	50	-	100	3
11.	CET-103	Engineering Graphics-I	-	-	3	3	50	50	-	100	4
12.	PHT-107	Physics-I (Pr)	-	-	2	2	-	60	40	100	3
13.	CHT-107	Chemistry-I (Pr)	-	-	2	2	-	60	40	100	3
14.	ELT-107	Basic Electrical Engineering.(Pr)	-	-	2/2	1	-	60	40	100	3
15.	COT-103	**OR** Computer (Pr) *	-	-	2/2	1	-	60	40	100	3
16.	MET-105	Workshop-I (Pr)	-	-	3	3	-	60	40	100	3
		TOTAL	15/17	8/7	12/11	35	575/600 200/160	225/240		1600	

- HUT-109 and (ECT-103 +ECT-105) will be offered to first half of the students strength.
- HUT-107 and MET-103 will be offered to second half of the students' strength.
- (ELT-105 + ELT-107) will be offered to first half of the students strength.
- (COT-101 + COT-103) will be offered to second half of the students strength.

* All engineering departments will share the teaching as well as the examinations.

Bachelor of Technology (Computer Engg.)
Scheme of Courses/Examination
(2nd SEMESTER)

(Common to all Branches)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)
			L	T	P/D	Tot	Th	Sess	P/VV	Tot	
1	MET-103	Manufacturing Processes **OR**	3	1	-	4	50	50	-	100	3
2	HUT-109	Engineering Economics	3	1	-	4	50	50	-	100	3
3	HUT-107	Commn. Skills in English **OR**	2	2	-	4	50	50	-	100	3
4	ECT-103	Basic Electronics Eng. and	2	1	-	3	50	50	-	100	3
5	ECT-105	Basic Electronics Eng.(Pr)	-	-	2/2	1		60	40	100	3
6	MAT-104	Mathematics-II	3	1	-	4	50	50	-	100	3
7	PHT-106	Physics-II **OR**	3	2/2	-	4	50	50	-	100	3
8	PHT-108	Physics-II	3	2/2	-	4	50	50	-	100	
9	CHT-106	Chemistry-II	2	1	-	3	50	50	-	100	3
10	COT-101	Computer Engineering * **OR**	2	2/2	-	3	50	50	-	100	3
11	ELT-105	Basic Electrical Eng.	2	2/2	-	3	50	50	-	100	3
12	MET-104	Engineering Graphics-II	-	-	6	6	50	50	-	100	4
13	PHT-110	Physics-II (Pr) **OR**	-	-	2/2	1	-	60	40	100	3
14	PHT-112	Physics-II (Pr)	-	-	2/2	1	-	60	40	100	3
15	CHT-108	Chemistry-II (Pr)	-	-	2	2	-	60	40	100	3
16	MET-106	Workshop-II (Pr)	-	-	3	3	-	60	40	100	3
17	COT-103	Computer (Pr) * **OR**	-	-	2/2	1	-	60	40	100	3
18	ELT-107	Basic Electrical Engineering (Pr)	-	-	2/2	1	-	60	40	100	3
		Total	14/15	7/7	14/13	35	575/575	270/310	155/115	1800	

- HUT-107 and MET-103 will be offered to first half of the students' strength..
- HUT-109 and (ECT-103 +ECT-105) will be offered to second half of the students strength.
- (COT-101 + COT-103) will be offered to first half of students strength.
- (ELT-105 + ELT-107) will be offered to second half of students strength.
- (PHT-108+PHT-112) will be offered to (C+M) students and (PHT-106+PHT-110) will be offered to other students.

* All engineering departments will share the teaching as well as the examinations.

Bachelor of Technology (Computer Engineering.)
Scheme of Courses/Examination
(3rd SEMESTER)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)	Credits
						Tot	Th	Sess	P/VV	Tot		
1	HUT-201	Industrial Sociology	2	1	-	3	60	40	-	100	3	2.5
2	COT-201	Programming Methodology and File Structures	3	1	-	4	60	40	-	100	3	3.5
3	COT-203	Data Structures	4	2	-	6	60	40	-	100	3	5.0
4	COT-205	Computer Organization & Architecture	4	1	-	5	60	40	-	100	3	4.5
5	COT-207	Discrete Structures	3	2	-	5	60	40	-	100	3	4.0
6	COT-209	Digital Electronics	3	2	-	5	60	40	-	100	3	4.0
7	COT-211	Programming Methodology (Pr)	-	-	2	2	-	60	40	100	3	1.0
8	COT-213	Data Structures (Pr)	-	-	3	3	-	60	40	100	3	1.5
9	COT-215	Digital System Design (Pr)	-	-	2	2	-	60	40	100	3	1.0
		Total	19	9	7	35				900		27

Bachelor of Technology (Computer Engineering)
Scheme of Courses/Examination
(4th SEMESTER)

Sl. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)	Credits
			L	T	P/D	Tot	Th	Sess	P/VV	Tot		
1	COT-202	Object Oriented Programming	3	1	-	4	60	40	-	100	3	3.5
2	COT-204	Programming Languages Concepts	3	1	-	4	60	40	-	100	3	3.5
3	COT-206	Software Engineering	3	1	-	4	60	40	-	100	3	3.5
4	COT-208	Microprocessors I	3	1	-	4	60	40	-	100	3	3.5
5	COT-210	Unix and Linux Programming	3	1	-	4	60	40	-	100	3	3.5
6	MAT-202	Mathematics III	3	1	-	4	60	40	-	100	3	3.5
7	COT-212	Object Oriented Programming (Pr)	-	-	2	2	-	60	40	100	3	1
8	COT-214	Microprocessors I (Pr)	-	-	2	2	-	60	40	100	3	1.0
9	COT-216	Software Engineering (Pr)	-	-	3	3	-	60	40	100	3	1.5
10	COT-218	Unix and Linux Programming (Pr)	-		3	3	-	60	40	100	3	1.5
		Total	18	7	9	34				1000		26

NOTE: Students of all branches will undergo a practical training of 6 weeks duration after the 4th semester exam.

Bachelor of Technology (Computer Engineering)
Scheme of Courses/Examination
(5th SEMESTER)
(w.e.f. July 2009)

Sl. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)	Credits
			L	T	P/D	Tot	Th	Sess	P/VV	Tot		
1	HUT-301	Business Management	3	1	-	4	60	40	-	100	3	3.5
2	COT-301	Design and Analysis of Algorithms	4	1	-	5	60	40	-	100	3	4.5
3	COT-303	Database Systems	3	2	-	5	60	40	-	100	3	4.0
4	COT-341	Analog and Digital Communication	4	1	-	5	60	40	-	100	3	4.5
5	COT-307	Automata Theory	4	2	-	6	60	40	-	100	3	5.0
6	COT-311	Algorithms Design (Pr)	-	-	3	3	-	60	40	100	3	1.5
7	COT-313	Database Systems (Pr)	-	-	3	3	-	60	40	100	3	1.5
8	COT- 315	Software Testing (Pr)	-	-	2	2	-	60	40	100	3	1.0
9	COT-317	Seminar	-	-	2	2	-		-	100	-	1.0
10	COT-319	Training Viva	-	-	-	-	-		-	100	-	3.0
Total			18	9	8	35				1000		29.5

Bachelor of Technology (Computer Engineering)
Scheme of Courses/Examination
(6th SEMESTER)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)	Credits
			L	T	P/D	Tot	Th	Sess	P/VV	Tot		
1	-	Departmental Elective I	3	2	-	5	60	40	-	100	3	4.0
2	COT-302	Operating Systems	4	2	-	6	60	40	-	100	3	5.0
3	COT-304	Computer Hardware Technologies	4	1	-	5	60	40	-	100	3	4.5
4	COT-306	Computer Networks	4	1	-	5	60	40	-	100	3	4.5
5	COT-308	Advanced Computer Architecture	4	1	-	5	60	40	-	100	3	4.5
6	COT-312	Operating Systems (Pr)	-	-	3	3	-	60	40	100	3	1.5
7	COT-314	Computer Hardware & Troubleshooting Lab (Pr)	-	-	3	3	-	60	40	100	3	1.5
8	COT-316	Advanced Technologies (Pr)	-	-	3	3	-	60	40	100	3	1.5
Total			19	7	9	35				800		27

NOTE: Students of all branches will undergo a practical training of 6 weeks duration after the 6th Semester exam.

Departmental Elective I

1. COT-340Digital Signal Processing
2. COT-342Multimedia Techniques
3. COT-344Graph Theory & Combinatorics
4. COT-346Advanced Database Systems
5. COT-348Logic of Programming
6. COT-350Computer Graphics
7. COT-352Software Quality and Reliability
8. COT-354Java Programming

Bachelor of Technology (Computer Engineering)
Scheme of Courses/Examination
(7^{th} SEMESTER)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)					Duration of Exam (Hours)	Credits
			L	T	P		Tot	Th	Sess	P/V V	Tot		
1		Open elective I	3	1	-		4	75	50	-	125	3	3.5
2		Departmental Elective II	3	1	-		4	75	50	-	125	3	3.5
3	COT-401	Internet and Intranet Engineering	3	1	-		4	100	25	-	125	3	4.5
4	COT-403	Microprocessors II	3	1	-		4	75	25	-	100	3	3.5
5	COT-405	Statistical Models for Computer Science	3	1	-		4	100	25	-	125	3	4..0
6	COT-411	Computer Networks Lab (Pr)	-	-	2		2	-	50	50	100	3	1.5
7	COT-413	Microprocessors II (Pr)			2		2	-	25	25	50	3	1.5
8	COT-415	Minor Project	-	-	5		5	-	50	75	125	3	6.0
9	COT-417	Seminar	-	1	-		1	-	50	-	50	-	1.5
10	COT-419	Training Viva	-	-	-		-	-	75	-	75	-	3.0
		Total	15	6	9		30				1000		32.5

Open Elective-I

1. COT-471 Fundamentals of Software Engineering (for branches except Co, IT)
2. COT-473 Fundamentals of Database Systems (for branches except Co, IT)
3. COT-475 Fundamentals of Computer Hardware Techniques (for branches except Co, IT)
4. COT-477 Artificial Intelligence (open to all branches)

Departmental Elective II

1. COT-441 Software Project Management
2. COT-443 Fuzzy Logic
3. COT-445 Parallel Computing
4. COT-447 Image Processing
5. COT-449 VLSI Technology
6. COT-479 Security and Cryptography

Bachelor of Technology (Computer Engineering)
Scheme of Courses/Examination

(8th SEMESTER)

Sr. No.	Course No.	Subject	Teaching Schedule				Examination Schedule (Marks)				Duration of Exam (Hours)	Credits
			L	T	P/D	Tot	Th	Sess	P/V V	Tot		
1	□□	Open Elective II	3	1	-	4	75	50	-	125	3	0
2	□□□	Departmental Electives III	3	1	-	4	75	50	-	125	3	4.0
3	COT-402	Web Engineering	3	1	-	4	100	50	-	150	3	4.5
4	COT-404	Compiler Design	3	1	-	4	100	25	-	125	3	4.5
5	COT-412	Web Engineering (Pr)	-	-	2	2	-	50	50	100	3	1.5
6	COT-414	Major Project	-	-	11	11	-	125	125	250	3	20
7	COT-416	Seminar	-	1	-	1	-	50	-	50	-	1.5
8	COT-418	General Fitness	-	-	-	-	-	-	-	75	-	3
Total			12	5	13	30				1000		39

□ **Open Electives II**

1. COT-472 Fundamentals of Operating Systems (for branches except Co, IT)
2. COT-474 Fundamentals of Computer Networks (for branches except Co, IT, EC)
3. COT-476 Object-Oriented Software Engineering (open to all branches)
4. COT-478 Expert Systems (open to all branches)
5. COT-480 Security & Cryptography (for branches Co, IT, EC)

□□ **Departmental Elective III**

1. COT-440 Distributed Operating Systems
2. COT-442 Software Project Measurement
3. COT-444 Natural Language Processing
4. COT-446 BioInformatics
5. COT-450 Software Testing

B.Tech I/II Semester
(Common to All branches)
Computer Engineering
COT-101

L	T
2	2/2

1. Introduction

Overview of a computer system, Block diagram and major parts of a computer, history of computer development, introduction to binary, octal, & hexadecimal numbers, ASCII code, different levels of programming languages – machine language, assembly language, high level language; need of operating system, tree structure of storage, introduction to assembler, compiler and interpreter.

2. Basics of C Language

C fundamentals; operators and expressions; i/p and o/p statements-getchar, putchar, scanf, printf; control statements – if-else, while, do-while, for, switch, break, continue, comma operator, goto statement.

3. Functions and Arrays

Defining and accessing a function, function prototype, passing arguments to a function; defining and processing an array, passing an array to functions, 2-d arrays, arrays and strings.

BOOKS

1. A. S. Tanenbaum : Structured Computer Organization, PHI. (Unit 1)
2. V. Rajaraman : Fundamentals of Computers, 3rd edition, PHI. (Unit 1)
3. Byron S. Gottfried : Theory and Problems of Programming with C Language, Schaum Series, TMH, 1998. (Unit 2, 3)
4. Kernighan and Richie : The C programming Language, 2nd Edition, PHI.(Ref. Book)

Course Outcome (COs)	Description
CO 1	Understand the fundamentals of digital and analog communication systems.
CO 2	Able to apply fourier analysis to communication signals and derive the power spectral density of signals.
CO 3	Able to define, formulate and analyze various techniques for amplitude and angle modulation.
CO 4	Analyse different techniques for digital data transmission and analyse the performance of spread spectrum communication systems.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	1			2		1	1		1		2
CO2	3	2	2	1	1			1				1
CO3	2	3	1		1	1	2	1	2	3	1	2
CO4	3	3	2	1	2	1		1	1	2		2

B.Tech I/II Semester
Computer Lab
COT-103

L	T	P
-	-	2/2

1. Understand the concept of operating system and learn related commands Write C programs for following :
2. Addition, subtraction, multiplication, division of 2 numbers
3. Find max and min of 3 numbers
4. Using while loop, find
   $S = 1 + 3 + 5 + \ldots$ upto N
   $S = x + x^2/2 + x^3/3 \ldots$ N terms
   Repeat these exercises using do-while loop.
5. Using for loop, calculate
   $S = x - x^3/3! + x^5/5! \ldots$ N terms
6. Using loops, print following design

   (a)
   ```
   1
   12
   123
   …N lines
   ```

   (b)
   ```
   *
   ***
   *****
   ... N lines
   ```
7. Read 2 numbers. Read the choice of operation. Add them if + is pressed. Subtract if – is pressed. Similarly for multiplication (*) and division (/).
8. Repeat exercise 7 such that program gets repeated again and again until user wants to exit.
9. Using function, compute ${}^{n}C_{m}$.
10. Using 1-d array read n numbers and find average. Also find the largest of these numbers. Use functions to implement these operations.
    Write modular programs for the following:
11. Implement following operations on matrices
    (a) Addition of two matrices (b) Transpose of a matrix
    (c) Multiplication of two matrices

Programming Methodology and File Structures
COT-201

L	T	P
3	1	-

1. Programming Methodology

Introduction to software development – Concept of professional programming and software development, tools for software design and for algorithm specification; Problem solving – Steps of developing any program, algorithm construction, patterns of logic; Use of sub-programs, Top-down decomposition, principles for decomposition, Communication between procedures, iteration vs. recursion; Program Documentation – Introduction, Documentation for programmers, operators, users, interactive systems.

2. Input/ Output Devices

Characteristics and working of magnetic tapes, magnetic disks – hard-disk, floppy disk, Winchester disks; optical disk, magnetic drum, magnetic bubble memory, charge-coupled device; overview of working of printers – character, line, page printers, scanners, plotters, OMR, MICR and Barcode Reader, Voice recognition and response devices.

3. File Systems and Organizations

File organizations, file operations, file systems, file directories, device control, channel and buffer management, Sequential File Organization – Creating, retrieving and updating sequential files & their performance; merging – natural, balanced, polyphase, cascade and their performance; relative file organization – definition, addressing techniques, problem of collision and its solutions, performance; indexed sequential file organization – definition, application, B+ tree structure, physical layout indexes, design; multi-key file organization – multi-key access, inverted file organization, multi-list file organization;

BOOKS

1. P. Juliff: Program Design, PHI, 2000.
2. M.E. Loomis: Data Management and File Structures, 2nd Ed. PHI. 1997.
3. V. Rajaraman: Fundamentals of Computers, PHI
4. S.K. Chauhan: P.C. Organization, Kataria & Sons, Delhi, 2001.
5. Alexis Leon and Mathews Leon: Introduction to Computers, Leon Press, Chennai & Vikas Publishing House, New Delhi, 2001.

Course Outcome (COs)	Description
CO 1	Able to distinguish between amateur and professional programming and develop professional approach to programming.
CO 2	Ability to handle large, complex problems through decomposition.
CO 3	Ability to select appropriate file organization and apply the appropriate merging techniques while external sorting, if needed.
CO 4	Able to understand the characteristics and working of various input and output devices of computer.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3	2	2	2				3	2	2	
CO 2	2	3	3	2	3						3	3
CO 3	3	2	1	2								1
CO 4	3	2	1	1	2		1					

Data Structures
COT-203

L	T	P
4	2	-

Note: Implementations are in C Language.

1. **Introduction**
   Fundamentals of pointers in C, pointer declaration, passing pointer to functions, pointers and arrays, dynamic memory allocation, Definition of Algorithm, Data Abstraction, Performance Analysis & Measurement, Files and related operations in C.
2. **Searching and Sorting Techniques**
   Searching techniques: Linear and Binary, Sorting techniques: Selection, Bubble, Insertion, Merge sort, Quicksort, List and Table Sorting.
3. **Simple Data Structures**
   Arrays: Definition of array, Array storage, sparse arrays; Transpose, addition, and multiplication of sparse matrices, Stacks and Queues and their applications, expression evaluation, A mazing problem; multiple stacks and queues in an array.
4. **Linked Data Structures**
   Linked Lists; definition, allocation for stacks and queues. Examples of linked lists, polynomial addition, comparison of sequential and linked allocation of storage; inversion, concatenation & copying of the lists. Implementations in C language.
   Doubly Linked List: Definition of circular and doubly linked list, header node, insertion and deletion, sparse matrix, representation using doubly linked lists. Examples for application of doubly linked lists; dynamic storage management; node structures, routines for allocation and deallocation, generalized lists and recursive algorithms for copying and comparison of lists.
5. **Advanced Data Structures**
   Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, Binary tree traversal techniques, Binary tree representation of trees, transformation of trees into binary trees, some more operations on binary trees, Binary Search Trees, Heaps and heapsort, Threaded binary trees, Graphs: Representation of graphs and their traversal, Minimum cost Spanning Trees.

BOOKS

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.B. Patel: Expert Data Structures in C, Khanna Publishers, 2001.
3. R.L. Kruse: Data Structures & Program Design in C, PHI.
4. D.F. Knuth: The art of Computer Programming Vol 1, Narosa Publications, 1985.
5. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum Series, TMH, 2005.

Course Outcome (COs)	Description
CO1	Apply the knowledge of various data structures and operations
CO2	To design various operations for solving programming real-world problems
CO3	To understand the concept of trees and graphs, their implementation and applications
CO4	Able to implement standard algorithms for searching and sorting
CO5	Analyze efficiency of different algorithms using time and space complexity

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3		2	2	2	1	1			1		3
CO2	3		2	1	2		1	1			1	1
CO3	2	2	2	3	1				1	1	1	2
CO4	2	2	2	1						1	1	1
CO5	3	2	2	3	1	2	1			1	1	2

Computer Organization and Architecture
COT-205

L	T	P
4	1	-

1. **Introduction**
   Basic Machine Principle, Structure and representation of real world data, Von-Newman Model and stored program concept, Subroutine, Branching & Macro facility.
2. **Processor Design**
   Processor Organization, Information representation and Number format, Instruction cycle and Instruction format, Addressing modes, Arithmetic operation, timed point addition, subtraction, multiplication and division, ALU design and floating point arithmetic, Parallel processing – Performance consideration, Pipeline processor and Multiunit processor.
3. **Control Design**
   Instruction sequencing and Interpretation, Hardware Control design method, Multiplier control unit and CPU control unit, Microprogrammed Control, Minimizing Instruction Size, Microprogrammed computer.
4. **Memory organization**
   Memory device characteristic, Random access and serial access memories, Virtual memory – memory hierarchies, Main Memory allocation & replacement policies, Segments, pages and file organization, High speed memories – Interlocked, cache and associative memory.
5. **System Organization**
   Local and long distance communication, Programmed I/O, DMA and interrupts, I/O processors & CPU – I/O interaction, Multiprocessor Introduction.

BOOKS

1. J.P. Hayes: Computer Architecture and Organization, 3rd Ed. TMH, 1999.
2. C.W. Gear: Computer organization and Programming, TMH.
3. T.C. Bartee: Digital Computer Fundamental, TMH.
4. M.M. Mano: Computer System Architecture, PHI.
5. A. S. Tanenbaum: Computer System Organization, PHI.

Course Outcome (CO's)	Description
CO1	Understand basic machine principles, concept of Newman Model and stored program.
CO2	Able to perform computer arithmetic operations, control unit operation and to analyze Pipeline processor and Multiunit processor.
CO3	Develop an ability to understand, how to organize system and ways to deal with interrupts.
CO4	Ability to access and organize various types of Memories and mapping process.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3	2	2	2				3	2	2	
CO 2	2	3	3	2	3						3	3
CO 3	3	2	1	2								1
CO 4	3	2	1	1	2		1					

Discrete Structures
COT 207

L T P
3 2 -

1. **Discrete Probability**:
   Introductory Examples, Basic definitions, Engineering applications of probability, Set theory, Sample space &events, Probability Multiplication principle, Product of sums principle, Cross product of Sample spaces, Theorem of Total Probability, Conditional Probability, Mutual Exclusion and Independent Events, Principle of Inclusion and Exclusion, Bayes' Rule.

2. **Discrete Random Variable & Distributions:**
   Random variables and their event spaces, probability Mass function, Distribution function, Discrete Uniform Distribution, Bernouli Trial & Binomial distribution, Poisson distribution, Geometric distribution, Mean & Variance of random variables.

3. **Relations and Algebraic system:**
   Binary Relation and their properties, Equivalence Relations and partitions, Partial ordering Relations, Functions and Pigeonhole Principle, Propositions, Definitions and elementary properties of algebraic structures , semi groups, monoids and submonoids, groups, and subgroups, Homomorphism and Isomorphism of monoids and Groups, Definition and Examples of Rings and Subrings, Types of Rings, Commutative Ring, Integral Domain, Division Ring, Relation of Isomorphism in the set of rings, Field, its characteristics and subfield.

4. **Graphs and Trees:**
   Introduction, Basic Terminology, Multigraphs and Weighted Graphs, Paths and Circuits, Shortest Paths in Weighted Graphs, Eulerian Paths and Circuits, Hamiltonian Paths and circuits, Planar Graphs, Trees, Rooted Trees, Path Lengths in Rooted Trees, Binary Search Trees, Spanning Trees and Cut-sets, Minimum spanning Trees.

BOOKS

1. C.L. Liu; Elements of Discrete Mathematics.
2. Discrete Mathematics for Computer Science, by Gary Haggard, J.Schlipf, S.Whitesides, Cengage Learning.
3. J.P. Tramblay; Discrete mathematical Structures with Applications to Computer Science, McGraw Hill, N.Y., 1977.

Course Outcome (COs)	Description
CO1	Recognize, identify, and solve problems using set theory and elementary number theory also applying the concepts of functions and relations in problem solving.
CO2	Apply appropriate mathematical structures (such as group, ring, domain) and their operations for solving/interpreting problems.
CO3	Formulate, represent and evaluate possible solutions to real world problems using the concepts of graph theory.
CO4	Establish results using various proof techniques and first order logic. Enumerate solutions using combinations and permutations

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	1	1		2						1
CO2	3	3	2	1		1						
CO3	3	3	2	1		2						1
CO4	3	3	1	2		2						2

DIGITAL ELECTRONICS
COT-209

L	T	P
3	1	-

1. **Number Systems and Codes**
   Introduction to positional number system, signed magnitude numbers, floating point numbers, binary arithmetic: addition, subtraction, multiplication and division, Base conversion, conversion formulas with examples, one's and two's compliment arithmetic,
   Computer codes – BCD codes, gray codes, excess-3 codes, parity checks, Hamming and alphanumeric codes.
2. **Digital Logic Families**
   Qualitative introduction to digital ICs, TTL, Schottky TTL, ECL, MOS Logic, CMOS Logic, Tri-state logic: Characteristics and properties.
3. **Combinational Logic Design**
   Introduction, standard representations for logical functions, Karnaugh map representation, simplification of logical functions using K-map, minimization of logical functions specified in minterms/maxterms or Truth Table, minimization of logical functions not specified in minterms/maxterms, Don't care conditions, design examples, Ex-or and Ex-nor simplification of K-maps, five and six-variable K-maps, QM method, MEV method.
4. **Combinational Logic Design using MSI circuits**
   Introduction, multiplexers and their use in combinational logic design, demultiplexers/decoders and their use in combinational logic design, adders and their use as subtractors, digital comparators, parity generators/checkers, code converters, priority encoders, 7-segment decoder/driver.
5. **Synchronous Sequential Circuits**
   Introduction, FSM model, memory elements and their excitation functions. Synthesis of synchronous sequential circuits, capabilities and limitation of FSM, state equivalence and minimization, simplification of incompletely specified machines.
6. **Asynchronous Sequential Circuits**
   Fundamental mode circuits synthesis, state assignment, pulse mode circuits.

BOOKS

1. R.P. Jain: Modern Digital Electronics, TMH.
2. Z Kohavi: Switching and Finite Automata Theory, TMH
3. M.M. Mano: Digital Logic Design, PHI.
4. Dr. B.R. Gupta: Digital Electronics, KATSON
5. James W. Bignell & Robert Donovan: Digital Electronics, ENGAGE LEARNING
6. Sanjay Kumar Bose: Digital Systems, NEW AGE INTERNATIONAL PUBLISHERS

Course Outcome(COs)	Description
CO1	To study the fundamentals of digital gates logic and number systems
CO2	Ability to design MSI circuits.
CO3	Knowledge about the function of Flip-Flops, Counters, Registers
CO4	Understanding the concepts of Sequential circuits.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	2		2					2	1	2
CO2		2	3	2		1			2			
CO3	2	2	2		2					2	2	2
CO4	1	3	3						1	1		

Programming Methodology (Pr.)
COT-211

L	T	P
-	-	2

Note: Implement following programs in C language.

1. Modular program development of a simple text based calculator.
2. Modify above design to develop scientific calculator.
3. Computation of ${}^{n}C_{m}$ using Recursion.
4. Generate Fibonacci series using recursion.
5. Implement natural merge and polyphase merge.
6. Implement a GUI/Mouse driven simple calculator.

Course Outcome (COs)	Description
CO1	Able to understand the concept of modular program development.
CO2	Design and development of recursion based program.
CO3	Design programs involving natural and polyphase merge.
CO4	Use different graphical based programming to create a simple calculator.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	2	2	2	1							1
CO2	2	1	2	1	1							1
CO3	2	2	2	2	2							
CO4	1	2	1	2	1							1

Data Structures (Pr.) (Using C language)
COT-213

L	T	P
-	-	3

Note: - *Write all programs in C language.*

1. Define two strings as arrays. Read them using %s. Using pointers, concatenate them without using string.h.
2. Define a pointer to an integer; read a list of n numbers using dynamic memory allocation and find average of these numbers.
3. Create a file containing 26 alphabets (A to Z) in separate lines.
4. Copy a file to another. Source file name and destination file name are input from the user.
5. Write a program for binary search (successful and unsuccessful both).
6. Sort n numbers using quick/merge/selection sort. Also count the number of exchanges in each case.
7. Write a program for expression evaluation using stacks.
8. Write a program for infix to postfix conversion.
9. Create a singly linked list and reverse it in the same list.
10. Write a program for a doubly linked list giving following option, insertion, deletion, retrieval,
11. Write a program to implement queues using linked list with option; list of elements in queue, insertion, and deletion.
12. Write a program to implement stacks using linked list with options push and pop.
13. Write a program for multiplication of two polynomials using linked list.
14. Write a program to implement binary trees. Depending on the choice, inorder/ preorder/ postorder traversal is done.
15. Implement heap sort. Show the contents of heap after each adjustment of element i.e. n outputs should be printed if list has n elements.

Course Outcome (COs)	Description
CO1	Able to performance analysis and measurement of memory allocation and file operations in C.
CO2	Implementation & Analysis of standard searching and sorting techniques.
CO3	Able to implement different data structures i.e. Array, Linked List etc.
CO4	Able to represent Tree and Graph Structure .Also implement the major tree & graph algorithms.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3		3	3		2	1		1	3	3
CO2	3	3		3	3		1			1	2	3
CO3	3	3		3	3		1			1	3	2
CO4	3	3		3	3					2	2	3

Digital System Design (Pr.)
COT-215

L	T	P
-	-	2

1. To study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, & EX-NOR).
2. To design and verify a half and full adder circuits.
3. To design a 4 bit adder/subtract using IC 7483.
4. To design and implement a 4:1 multiplexer.
5. To design and implement a 1:4 demultiplexer.
6. Verify the truth table of a 4-bit comparator using IC 7485.
7. To design and verify a 2:4 decoder.
8. To design and implement a 2:4 encoder.
9. To verify the operation of a D and JK flip-flop using ICs 7474 AND 7473.
10. To design and verify the operation of RS, T, D, and JK flip-flops using logic gates.
11. To verify the operation of a Mod-10 counter.
12. To design and implement the operation of a Mod-16 counter using JK flip-flops
13. To design and implement a Mod-10 counter using JK flip flops and logic gates.
14. To verify the operation of a 4 bit shift register using IC 7495.
15. To design and verify the operation of a 4-bit shift left register using D flip-flops.
16. To design and verify the operation of a 4-bit shift right register using D flip-flops.

Course Outcome (COs)	Description
CO1	Verify the operation of logic gates
CO2	Ability to design the Combinational circuits.
CO3	Analyze the operation of Flip-Flops, Counters and Registers by designs.
CO4	Able to apply the knowledge of finite state machines to design sequential circuits.
CO5	Grasp an understanding of Multiplexers and Demultiplexers and their use in combinational logic design.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	2	2	2	1							1
CO2	2	1	2	1	1							
CO3	2	2	2	2	2							1
CO4	2	2	2	2	1							1
CO5	2	2	1	2	2							1

Object Oriented Programming
COT–202

L	T	P
3	1	-

1. **Object Oriented Programming and Design**
   Review of Abstraction, Objects and other basics, Encapsulation, Information hiding, Method, Signature, Classes and Instances, Polymorphism, Inheritance, Exceptions and Exception Handling with reference to object modeling, Object Oriented Design – Process, Exploration and Analysis.
2. **C++ Programming Basics**
   Fundamentals: Variables and assignments, Input and Output, Data Types and Expressions, Flow of control, Subprograms: Top down design, Predefined functions, Programmer defined functions, Procedural abstractions, Local variables, Overloading function names, Parameter passing, Structures for diverse data, Structures as function arguments, Initializing structures, Defining classes and member functions, Public and private members ,Constructors for initializations, , Destructors, Copy constructors, Friend functions.
3. **C++ Object Oriented Concepts**
   Objects and Classes: Operator overloading , Overloading the assignment operator ,subscript operator , this pointer ,Use of file for I/O, Formatting output with stream functions, Character I/O, Inheritance, constructors in inheritance , Standard C++ classes, Derived classes ,Virtual functions, virtual base class.
4. **C++ Data Structures and Advanced Topics**
   Arrays – Programming with arrays, arrays of classes, arrays as function arguments, Strings, Multidimensional arrays, Arrays of strings, Pointers, Dynamic arrays, Classes and dynamic arrays, Templates
   – generic classes and functions, namespaces, introduction to STL.
5. **Introduction to Java**
   Data types, Variables and Assignment, String and Characters, Arrays, Control statements, Loops, Operators. Introduction to Classes, Constructors, this keyword, Static, Local and
   Instance variables, Methods, Method overloading, Method overriding, subclasses, inheritance, modifiers, polymorphism.

BOOKS

1. Herb Schildt: C++ - The Complete Reference, TMH, Delhi
2. R.Venugopal :Mastering C++,TMH,Delhi
3. Bruce Eckel : Thinking in C++ Volume I ,Pearson Education,Delhi.
4. Horstmann: Computing Concepts with C++ Essentials, John Wiley.
5. Joseph O'Neil and Herb Schildt: Teach Yourself JAVA, TMH, Delhi.
6. Gary Cornell , Sun Microsystems: Core Java 2,Pearson Education, Delhi

Course Outcome (COs)	Description
CO 1	Able to differentiate between structure oriented programming and object oriented programming and to specify simple abstract data types and design implementations
CO 2	Recognize features of object oriented design such as encapsulation, polymorphism, inheritance and composition of systems based on object identity.
CO 3	Able to use object oriented programming language like C++ and associated library to develop object oriented programs.
CO 4	Able to understand and to use basic fundamentals of Java.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3	3		1	1			1	1		1
CO 2	3		2	1	1	1					1	1
CO 3	3	3	1	3					1	1		
CO 4				2	3					1		1

Programming Languages Concepts
COT-204

L	T	P
3	1	-

1. **Describing Syntax and Semantics**
   Introduction, general problem of describing syntax, formal methods of describing syntax, attribute grammar dynamic semantics.
2. **Data Types**
   Name, variables, binding, types of binding, type checking, strong typing, type conversion, named constant, principle data types, character string, user defined data types, pointer and reference. Discuss with reference to C, C++, and JAVA.
3. **Subprogram**
   Fundamentals of subprogram, referencing, environment – local and non local parameter passing, subprogram name as parameter, overloaded subprogram, generic subprograms, coroutine, CALL-RETURN structure, recession, implementing non-local referencing environment, scope-static and dynamic, implementation of scopes. Discuss with reference to C, C++, and JAVA.
4. **Concurrency and Exception Handling**
   Introduction, subprogram level concurrency, and synchronization, through semaphores, monitors and message passing, Introduction to Exception handling. Discuss with reference to C, C++, and JAVA.
5. **Storage Management**
   Major routine storage elements, programmer and system controlled storage management, storage management phases, static storage management, stock based storage management, heap storage management –fixed size and variable size.
6. **Functional Programming & Logical Programming**
   Fundamentals of functional programming, LISP: data types, functions, control flow, applications, Overview of logical programming, basic elements of Prolog, deficiencies and application of Prolog.

BOOKS

1. R. W. Sebesta, Concepts of Programming Languages (Addison-Wesley Pub).
2. T.W. Pratt, Programming Languages: Design & Implementation, PHI, 3rd Ed.
3. E. Horowitz, Fundamentals of Programming Languages.

Course Outcome (COs)	Description
CO1	Able to express the computational solutions of programming language
CO2	Know and understand the various data types, subprograms with reference to C, C++ , JAVA
CO3	Be familiar with the concepts of concurrency, exception handling and storage management
CO4	Detail study of functional programming and logical programming

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	2	3	1					2	3	2
CO2	3	2	2	2	2					2	3	2
CO3	2	2	2	2	2					2	2	2
CO4	2	2	2	1	1					1	2	2

Software Engineering
COT-206

L	T	P
3	1	-

1. **Introduction**
   Introduction to Software crisis & Software processes; Software life cycle models – Build & Fix, waterfall prototype evolutionary, spiral model.
2. **Requirement Analysis & Specifications**
   Problem Analysis – DFD, Data dictionaries, ER diagrams, object diagrams; approaches to problems analysis; SRS; specifying behavioral & non-behavioral requirements.
3. **Software Design**
   What is design? Modularity, strategy of design, function oriented design, object oriented design.
4. **Software Metrics**
   Introduction, size metrics, data structure metrics, information flow metrics, entropy-based measures, metric analysis.
5. **Software Reliability**
   Importance, Software reliability & Hardware reliability, failures & faults, reliability concepts, reliability models – macro, basic, logarithmic Poisson, calendar time component, micro models; estimating number of residual errors; reliability allocation.
6. **Software Testing**
   Introduction, Functional testing, structural testing, activities during testing, debugging, testing tools.
7. **Software Maintenance**
   Introduction, types of maintenance, maintenance process, maintenance models, reverse engineering, re-engineering.

BOOKS

1. K.K.Aggarwal, Yogesh Singh: Software Engineering, New Age International Ltd, 2001.
2. R.S. Pressman, Software Engineering – A Practitioner's Approach, 5th Ed, TMH, 2000.
3. Ian Sommerville, Software Engineering, 4th Ed., Addison Wesley.
4. Pankaj Jalote, An Integrated Approach to Software Engineering 2nd Ed, Narosa Publishing.

Course Outcomes (COs)	Description
CO 1	Able to apply the concepts to choose an appropriate software process models according to given user requirements
CO 2	To analyze requirement techniques like Data flow diagram, Entity relationship diagram, Object diagram
CO 3	Understanding the concept of Software Design and various strategies involved in software design and emphasizing upon various software metrics used for analyzing the software
CO 4	Design various software reliability measures to assess the quality of software in case of various faults and failures.
CO 5	Develop various testing methodologies debugging tools and maintenance models to ensure the accountability of software.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1	2		3	2		1	2	2	3	1	2
CO 2	1	2	2	2	2	1		1	2	2		2
CO 3		2	2		2		1		2	2	2	3
CO 4	2		2	2	2	2	1	2	2	2	1	2
CO 5	1		2		3	1	1	2	3	3		2

Microprocessors - I
COT-208

L	T	P
3	1	-

1. 8086 Architecture
CPU architecture, internal operation, machine language instructions, instruction execution time.

2. Assembly Language Programming
Assembler, instruction format, data transfer instruction, arithmetic instructions, branch instruction, NOP & HLT instructions, flag manipulation instruction, logical instruction, shift and rotate instruction, directions and operators.

3. Modular Programming
Stacks, Procedures, Basic Interrupt processing, Interrupt and interrupt routines, 8259A Programmable Interrupt Controller, macros – local labels and nested macros.

4. Strings and I/O Programming
String instructions, I/O consideration, programmed I/O block transfer and DMA.

5. I/O Interface
Serial communication, asynchronous, synchronous, physical, 8251A; Parallel communication: 8255 A, DMA controllers; maximum mode, 16-bit bus interface, 8279 Programmable keyboard/ Display interface, 8254 Programmable interval Timer, Interfacing to A/D and D/A converters, Stepper motor interfacing.

BOOKS

1. Liu and Gibson, Microcomputer Systems: 8086/8088 family: Architecture, Programming and Design, PHI.
2. D.V. Hall, Microprocessors and Interfacing, TMH.
3. A. K. Ray & K.M. Bhurchandi, Advanced Microprocessors and Peripherals: Architecture, Programming and interfacing, TMH.
4. Bray, The intel Microprocessor 8086/8088-Pentium: Architecture, Programming and interfacing , PHI.
5. James L. Antonakos, The intel Microprocessor family H/W and S/W Principles and Applications, Cengage Learning.

Course Outcome (COs)	Description
CO 1	Develop an understanding of basic concepts of microprocessors (4004 to Pentium-IV)
CO 2	Understand the instruction set of 8086
CO 3	Able to apply the knowledge of assembly language to solve various problems
CO 4	Grasp an understanding of various peripheral device interfaces with 8086
CO 5	Able to design and implement various interfaces in real life different applications.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3			1	3	1		1	1		1
CO 2	2	2	3	2	2		1		1	2		2
CO 3	1	3		3	3	3	1	2	2	1	2	3
CO 4	2	2	3	2	3		1	1	2	2	2	3
CO 5	1	2	2	3	3		1	1	3	1	2	3

Unix and Linux Programming
COT-210

L	T	P
3	1	-

1. **Linux Startup**
   User accounts, accessing linux – starting and shutting processes, Logging in and Logging out, Command line, simple commands

2. **Shell Programming**
   Unix file system: Linux/Unix files, inodes and structure and file system related commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing environment

3. **Regular Expressions and Filters**
   Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to egrep, sed, programming with awk and perl.

4. **The C Environment**
   The C compiler, vi editor, compiler options, managing projects, memory management, use of makefiles, dependency calculations, memory management – dynamic and static memory, building and using static and dynamic libraries, using ldd, soname, dynamic loader, debugging with gdb

5. **Processes in Linux**
   Processes, starting and stopping processes, initialization processes, rc and init files, job control – at, batch, cron, time, network files, security, privileges, authentication, password administration, archiving, Signals and signal handlers, Linux I/O system

BOOKS

1. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2000.
2. Sumitabha Das: Your Unix – The Ultimate Guide, TMH, 2000.
3. Mathew: Professional Linux Programming, vol.1 & 2, Wrox-Shroff, 2001.
4. Welsh & Kaufmann: Running Linux, O'Reiley & Associates, 2000.

Course Outcome (COs)	Description
CO 1	Understanding of Unix/Linux operating system, learn the installation procedure and basic commands.
CO 2	Knowledge of Unix files system/file handling, command line interpreter and shell programming.
CO 3	Ability to handle the files or records through filters by using regular expression.
CO 4	Analysis of C environment, libraries, memory management and debugging methodologies.
CO 5	Understanding of process manipulation and learning password management for user privileged in Linux.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2			2		1	1	1				
CO 2	3	3		1								
CO 3	3	3	2	3								
CO 4	2	3	2	2	3							
CO 5	1				2			3	3			

Object Oriented Programming (Pr.)
COT-212

L	T	P
-	-	2

Note:- Write programs first in C++ and later in Java.

1. (a) Model a geometric point to find distance between two points.
   (b) Model complex numbers and their operations.

2. Describe a class called TOLL- BOOTH with the following data items unsigned int - to hold the number of cars passing through the booth, double - to hold the total amount collected.
   Include the following member functions:
   * a constructor that sets both the data fields to zero.
   *PAYINGCAR( ) that increases the numbers of cars by one and increase the total amount by 2.50.
   *NOPAYING( ) that increases the number of cars but keeps the total amount unchanged.
   *DISPLAY( ) that displays both the total number of cars passing and the total number of amount collected.
   Write main( ) to test the class thoroughly.

3. Create a class rational which represents a numerical value by two double values- NUMERATOR and DENOMIATOR . Include the following public member functions:

* constructor with no arguments (default)
* constructor with two arguments.
* void reduce() that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
* overload + operator to add two rational numbers.
* overload >> operator to enable input through cin.
* overload << operator to enable output through cout.

Write a main () to test all the functions in the class .

4. Consider the following class definition class father {

```
protected : int age;

public;
father (int x){age =x;}
virtual void iam()
{cout <<"I AM THE FATHER, my age is :"<<age<<end1;}
};
```

Derive the two classes son and daughter from the above class and for each, define iam() to write out similar but appropriate messages. You should also define suitable constructors for these classes.

Now, write a main () that creates objects of the three classes and then calls iam() for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam() through the pointer to demonstrate polymorphism in action.

5. A thermostat is a device that keeps a system at a constants temperature. It behaves like a temperature gauge that is capable of getting the current temperature from the system. It is also a switch that can be turned "on" and "off". The thermostat monitors the temp. in the following manner:

   if the current temp. falls below 95% of the required temp., it turns itself "on". On the other hand , if the current temp. exceeds 1.05 of the required temp. ,it turns itself "off" .In all other cases ,its on-off status remain un changed.

   Implement classes for temp. gauge and switch(named switch) with suitable data and member functions. The temp. gauge class must have a member function get_temp() that will pretend to get the current temp. of the system by actually reading it from the keyboard.

   Now, implement thermostat class in both the following ways:

a) Develop a class called thermostat that include objects of temp. gauge and switch as its member(aggregation).

b) Develop a class called thermostat that inherits the data methods of temp. gauge and switch(multiple inheritance).

Write main() to test all the features of above mentioned classes.

6. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

7. Using the file created in problem 6, write a program to display the roll no. and names of the students who have passed (has obtained 50 or more) .

8. You are to create a file containing n records. Each record relates to a historical event and the year in which the event took place

   Some examples are:

   India Wins Freedom 1947

   Amartya Sen Gets Nobel 1998

   First World War Begins 1914

   The data should be read from terminal while creating the file.

9. A hospital wants to create a database regarding its indoor patients. The information to store include
   (a) Name of the patient
   (b) Date of admission
   (c) Disease
   (d) Date of discharge

   Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the databases. Create a derived class to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

10. Define a class to store the time at a point. The data members should include hr., min., and sec. to store hours, minutes and seconds. The member functions should include functions for reading the time and displaying the same. Add a friend function to add two times. Write a program, using the above declaration, to read two times and add them.
11. Write a program to read two matrices and find their product. Use operator overloading so that the statement for multiplying the matrices may be written as
    $Z = X * Y$ where X , Y and Z are matrices.
12. Write a program to read a number and display its square, square root, cube and cube root. Use a virtual function to display any one of the above.
13. Make a class **Employee** with a name and salary. Make a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type String. Supply a method to **toString** that prints the manager's name, department and salary. Make a class **Executive** inherit from **Manager**. Supply a method **toString** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.
14. Write a superclass **Worker** and subclass **HourlyWorker** and **SalariedWorker**. Every worker has a name and a salary rate. Write a method **computePay(int hours)** that computes the weekly pay for every worker. An hourly worker gets paid the hourly wage for the actual number of hours worked, if **hours** is at most 40. If the hourly worker worked more than 40 hours, the excess is paid at time and a half. The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is. Write a static method that uses polymorphism to compute the pay of any **Worker.** Supply a test program that tests these classes and methods.

BOOKS

1. Computing Concepts with C++ Essentials by Horstmann, John Wiley.
2. C++: The Complete Reference by Schildt, TMH, Delhi.
3. Java2: The Complete Reference by Schildt, TMH, Delhi.
4. Computing Concepts with Java 2 Essentials by Horstmann, John Wiley.
5. Teach Yourself Java by Joseph O'Neil (TMH, Delhi).

Course Outcome (COs)	Description
CO 1	To understand object oriented approach
CO 2	To reuse the code and write the classes which work like built-in types
CO 3	To demonstrate the concept of object oriented via debug, maintain and extend
CO 4	To be able to apply object-oriented concepts in real world applications

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	3	2		2							1
CO 2	1	2	2	3		1					1	
CO 3		3	3		3					1	1	
CO 4	2	2	1	2	2	1			1	1		1

Microprocessors – I (Pr.)
COT-214

L	T	P
-	-	2

1. Write a program to print the alphabets.
2. Write a program to read a integer number of max (16 bit), store that number in a register and display it digit by digit.
3. Repeat exercise 2 for 32bit number.
4. Write a program to find factorial of a number, where result does not exceed 32 bit. Use procedure to calculate factorial and pass parameters
5. Write modular program to perform addition, subtraction, multiplication and division of two 16-bit numbers.
6. Repeat exercise 5 for two 32-bit numbers
7. Sort n numbers using modular program.
8. Check whether a given string is palindrome or not.
9. Reverse an input string.
10. Merge two sorted list of integers
11. Write a program to print the date of BIOS.
12. Write some programs, which use multiple data segments and multiple code segments. Do these programs by defining different segments in different files and link all of them to get the desired output.
13. Using INT 10h, change the size of cursor, change the position of the cursor based on user's choice.
14. Use INT 10h, change the mode of monitor and draw a single pixel in each color available in video mode 13.
15. Use INT 10h, draw a line in graphics mode.
16. Use INT 33h, write a program which turns on your mouse cursor and Pause the computer until a mouse button is pressed.
17. Use INT 33h, write a program to create a textual mouse "button".
18. Use INT 33h, write a program to check for the presence of a mouse driver.

Course Outcome (COs)	Description
CO 1	Develop a basic understanding of syntax of instruction set of 8086.
CO 2	Able to apply the knowledge of assembly language to solve various problems.
CO 3	Grasp an understanding of various peripheral device interfaces with 8086.
CO 4	Develop the knowledge to implement various interfaces through assembly language programming.
CO 5	Apply knowledge to implement different applications like stepper motor, traffic light, temperature monitoring etc.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1	3	1	1	1		1		1			
CO 2	2	3	3	2	3		2		1	1	2	1
CO 3	1	3		3	1	2	3	1	2	1	2	1
CO 4	2	2	3	2	3	3	2	1	3	1	3	2
CO 5	1	2	2	3	3	3	3	1	3	1	3	2

Software Engineering (Pr.)
COT-216

L	T	P
-	-	3

Note:- Implement the following programs using C.

1. Implement Halstead's equation to compute various science metrics like volume etc., language level, estimated program length, effort and time in a program.
2. Compute average number of live variables per statement in a program.
3. Compute average life of variables in a program.
4. Compute psychological complexity of a program.
5. Compute McCabe's cyclomatic complexity of a program and generate its control graph.
6. Use some CASE tool for identifying various phases of software engineering, generate SRS document, design document like DFD and ER diagram, test cases generation for result automation, engineering admission automation (seat allocation during counseling).

Course Outcome (COs)	Description
CO 1	Implement Halstead's equation to compute software metric.
CO 2	Computation of psychology and McCabe's complexity of program.
CO 3	Develop an ability to understand case tool and its use in software engineering phases.
CO 4	Computation of average variable in program.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	2	2	1					1		3
CO2	3	2	1	2	1		1		1	1		3
CO3	1			2	3	2	2	1	2	3	3	3
CO4	2	1	2	2		1			1			2

Unix and Linux Programming (Pr.)
COT-218

L	T	P
-	-	3

1. Familiarize with Unix/Linux logging/logout and simple commands.
2. Familiarize with vi editor and Linux GUIs.
3. Using Bash shell develop simple shell programs.
4. Develop advanced shell programs using awk and grep.
5. Compile and debug various C programs using different options.
6. Learning of installation and upgradation of Linux operating system.
7. Install Linux on a PC having some other previously installed operating system. All OSs should be usable.
8. As supervisor create and maintain user accounts, learn package installation, taking backups, creation of scripts for file and user management, creation of startup and shutdown scripts using at, cron etc.

Course Outcome (COs)	Description
CO 1	Ability to install the Linux operating system, hands on with simple commands, analysis of vi editors, environment variable settings etc.
CO 2	Able to develop a C program. Apply the concepts of file handling, filters, and regular expression using shell programming.
CO 3	Capable to develop and implement grep, awk, and perl scripts.
CO 4	Ability to develop shell script program that handle processes.
CO 5	Apply the concepts of debugging and execute shell programming. Installation of Linux package into the operating system.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1			2							3	
CO 2	3	3	3	2								
CO 3	1	3	1	2								
CO 4	2	3	2	2					3	2		
CO 5		2		3				2	1		3	

Design and Analysis of Algorithms
COT-301

L	T	P
4	1	-

1. **Introduction**
   Review of elementary data structures, analyzing algorithms, asymptotic notation, recurrence relations, Hash tables, Binary search trees, Convex Hull problem.
2. **Sorting and Order Statistics**
   Heapsort, Priority queues, Quicksort, Sorting in linear time.
2. **Advanced Design and Analysis Techniques**
   Dynamic programming – Elements, Matrix-chain multiplication, longest common subsequence, Travelling Salesperson problem, Greedy algorithms – Elements, activity-selection problem, Huffman codes, task scheduling problem, Knapsack Problem, Backtracking – Elements, 8 – Queens, Graph Coloring, Hamiltonian Cycles.
3. **Advanced Data Structures**
   Operations in B-Trees, Binomial heaps, Fibonacci heaps, data structures for disjoint sets, strings.
4. **Graph Algorithms**
   Review of graph algorithms, topological sort, strongly connected components, minimum spanning trees – Kruskal and Prim's, Single source shortest paths, relaxation, Dijkstra's algorithm, Bellman-Ford algorithm, single source shortest paths for directed acyclic graphs.
5. **NP – Hard & NP – Complete problems**
   Basic concepts, Clique Decision problem, Node Cover decision problem, Travelling Salesperson decision problem, Introduction to approximation algorithms Planer Graph Coloring, Maximum programs stored problem.

BOOKS

1. Cormen, Leiserson and Rivest: Introduction to Algorithms, 2/e, PHI.
2. Horowitz, Sahni, and Rajasekaran: Fundamentals of Computer Algorithms, Second Edition, Universities Press, Hyderabad.
3. Aho, Hopcroft, and Ullman: The Design and Analysis of Computer Algorithms, Addison Wesley.

Course Outcome (COs)	Description
CO1	Gain insight about design and analysis of standard searching and sorting algorithms. Learn various algorithm Analysis techniques.
CO2	Learn divide and conquer, Dynamic programming, Greedy and backtracking paradigms and understand when an algorithmic design situation calls for them.
CO3	Able to compare between different data structures i.e., trees, heaps etc. also, pick an appropriate data structure for a design situation.
CO4	Explain the major graph algorithms and their analysis. Employ graphs to model engineering problems.
CO5	Understand NP completeness and understand difference between NP-Hard & NP-complete problems.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	2	2	2	1	1	1	1		3	2
CO2	3	3	3	3	2	1	2	1			3	2
CO3	2	3	3	2	2	1			1	1	2	1
CO4	3	3	3	3	3	1					2	
CO5	2	2	2	3	2	1					3	

Database Systems
COT-303

1. **Basic Concepts**
   What is database system, why database, Data independence, 3 levels of architecture; external level, conceptual level, internal level, mapping DBA, DBMS, organization of databases, components of DBMS, Data Models, Relational Models, Networks data model, Hierarchical Model, semantic data model.

2. **Relational Model**
   Introduction – Relational Model, base tables & views, relations, domains, candidate keys, primary key, alternate keys, foreign key, Integrity rules, relational Operators – relational algebra, relational calculus, Data Base Design – Introduction, Basic Definitions, Non-loss decomposition and functional dependencies, 1NF, 2NF, 3NF, BCNF, MVD & 4NF, JD & 5NF, Normalization procedure, other normal forms.

3. **Transaction Management**
   Transaction concept, transaction states, state diagram of a transaction, concurrent execution of transactions, conflict serialzability, view serializability, recoverable schedule, cascadeless schedule, testing of serializabity.

4. **Concurrency control**
   Lock based protocols, timestamp based protocols, and validation based protocols.

BOOKS

1. Database System Concept by Avi Silberchatz, Henry F. Korth and S. Sudarshan, Fifth Edition McGraw-Hill.
2. Fundamental of Database Systems by Ramez Elmasri, Shamkant B. Navathe, 5th Edition, Perason Education.
3. Database Management Systems, by Raghu Ramakrishnan, 3rd edition, MGH.

Course Outcome (COs)	Description
CO 1	Able to understand the basic concepts, principles and applications of database system.
CO 2	Able to discuss the components of DBMS, data models, Relational models.
CO 3	Able to use knowledge to find the functional dependencies and differentiate between different normal forms.
CO 4	Able to execute transaction concepts.
CO 5	Able to implement concurrency control protocols.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3		2	2	2	1	1			1	3	
CO 2	3		2	1	2		1	1			1	
CO 3	2	2	2	3	1				1	1	2	
CO 4	2	2	2	1						1	1	
CO 5	3	2	2	3	1	2	1			1	2	

Analog and Digital Communication
COT-341

L	T	P
4	1	-

1. **Spectral Analysis**
   Fourier series, Response of linear system Power spectral density, Fourier Transform, Convolution, Parseval's Theorem, correlation between waveforms, Impulse Function, ideal low pass filter, Hilbert transform, Pre-envelope.
2. **Random Variables & Noise**
   Probability, Joint Probability, Random variables, Cumulative distribution function, Probability density function, Average value of random variables, Properties of rectangular Gaussian, Binomial Rayleigh & Poisson's distribution, Central Limit Theorem, Error function, Random processes, Ergodicity & Stationarity, Covariance, spectral densities. Noise and its sources, Methods of noise calculation in network and interconnected networks, Mathematical representation of random noise. Narrow band noise and its representation, Transmission of noise through linear systems, Noise figure to noise temperature, Computation of signals to noise ratio. Noise bandwidth.
3. **Analog Modulation Techniques**
   Introduction, Amplitude Modulation, AM demodulators, Spectrum of AM signal, Double sideband suppressed carrier modulation, single side band modulation, Methods of generating SSB signals, vertical side band modulation, frequency division multiplexing; Angle modulation, Phase and frequency modulation, spectrum of F.M. signal bandwidth of FM signal; Effect of modulation index on bandwidth; NBFM & WBFM FM generation methods, FM Demodulation methods.
4. **Pulse Modulation**
   Sampling theorem for low pass and band pass signals, time division multiplexing, concept of pulse amplitude modulation and pulse width modulation, demodulation of signals, pulse code modulation, Noise computation in PCM systems, Delta modulation and adaptive delta modulation.
5. **Digital Modulation Techniques**
   Binary phase shift keying, differential phase shift keying, quadrature phase shift keying, M-ary PSK, QASK, Binary FSK, M-ary FSK, Minimum shift keying.
6. **Spread Spectrum Signals**
   Spread spectrum model, direct sequence spread signals, generation of PN sequences, CDMA system based on frequency hopped spread spectrum signal.

BOOKS

1. Taub and Schilling: Principles of Communication System, TMH.
2. Simon Haykin: Digital Communication, John Wiley.
3. J. G. Proakis: Digital Communications, MGH.
4. G. Kennedy: Electronic Communication System, TMH.

Course Outcome (COs)	Description
CO 1	Understand the fundamentals of digital and analog communication systems.
CO 2	Able to apply fourier analysis to communication signals and derive the power spectral density of signals.
CO 3	Able to define, formulate and analyze various techniques for amplitude and angle modulation.
CO 4	Analyse different techniques for digital data transmission and analyze the performance of spread spectrum communication systems.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	1			2		1	1		1		2
CO2	3	2	2	1	1			1				1
CO3	2	3	1		1	1	2	1	2	3	1	2
CO4	3	3	2	1	2	1		1	1	2		2

Automata Theory
COT-307

L	T	P
4	2	-

1. Introduction

Introduction to Finite State Machine, Binary counter, parity bit generator, Moore and Mealy FSMs, Equivalence, Isomorphism, Reduction of States, Regular Languages, Regular expressions, The memory required to recognize a language, Distinguishing one string from another, unions, Intersections and Complements, NFA, NFA with – transitions, Criterion for Regularity, Minimal Finite Automata, The pumping lemma, decision, problems, Finite automata, Nondeterminism and Kleen's Theorem, Regular and Non-regular languages.

2. Context-Free Language

Context – Free Grammars, Definition of CFG, example of familiar languages, unions, concatenations and closures of CFLs, Derivation Tree, Ambiguity, unambiguous CFG for algebraic expressions, Simplified forms and normal forms. Push down automata, definition, deterministic PDA, PDA to CFG and Vice Versa, Parsing. Context Free and Non Context Free Languages, Pumping lemma for CFG, Intersection and complements of CFL.

3. Turing Machines

Definition, Turing Machining as Language acceptors, combining TM, computing Partial Function with TM. Recursively Enumerable and Recursive Languages, Halting Problem, Post's correspondence Problem, Regular Grammars, context Sensitive grammars, Chomsey Hierarchy.

4. Computability

Primitive Recursive Functions, Primitive Recursive Predicates and some bounded operations, unbounded minimization and recursive functions, Godel Numbering, Non-numeric-functions. Growth rates of functions, Time and space complexity of TM, complexity Classes. P and NP. Polynomial-Time. Reductions and NP-Completeness, Cook's Theorem.

BOOKS

1. John C. Martin: Introduction to Languages and the Theory of Computation, MGH.
2. Lewis & Papadimitriou: Elements of the Theory of Computation, PHI.
3. Daniel I.A. Cohen: Introduction to Computer Theory: John Wiley.
4. J.E. Hopcroft and J.D. Ullman: Introduction to Automata Theory Languages and Computation, Narosa.

Course Outcome (COs)	Description
CO 1	Gain insight of theoretical foundations of computer science and be familiar with finite automata (deterministic or non-deterministic), FSM and master regular languages.
CO 2	Be exposed to types of languages (type-0, type-1, type-2, and type3), context-free languages (type-2) and using pushdown automata (PDA) as its acceptor.
CO 3	Designing of turing machines for various mathematical function, string function etc.
CO 4	Understand concept of decidability, unsolvability and recursive enumerability.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	3	3	3	2	1	1			1		2
CO2	2	3	3	3	2	2	2		1	1		2
CO3	2	2	2	2	2	2	2					2
CO4	1	1	1	1	1	1	1			1		1

Algorithms Design (Pr.)
COT-311

L	T	P
-	-	3

1. Implement the minimum cost spanning tree algorithm.
2. Implement the shortest path algorithm.
3. Implement the algorithm to compute roots of optimal subtrees.
4. An Euler circuit for an undirected graph is a path that starts and ends at the same vertex and uses each edge exactly once. A connected undirected graph G has an Euler circuit. If and only If energy vertex is of even degree. Give an algorithm and implement to find the Euler Circuit in a graph with e edges provided one exists.
5. Give an algorithm to determine whether a directed graph with positive and negative cost edges has negative cost cycle.
6. Write an algorithm in which given an n x n matrix M of positive integers is given and that finds a sequence of adjacent entries starting from M[n,1] and ending at M[1,n] such that the sum of the absolute values of differences between adjacent entries is minimized. Two entries M[i,j] and M[k,l] are adjacent if
   (a) i=K+_ 1 and j = l, or
   (b) i= k and j = l+_1

For ex. in the following fig. Sequence, 7,5,8,7,9,6,12 is a solution.

1	9	6	12
8	7	3	5
5	9	11	4
7	3	2	6

Matrix of Positive Integers.

7. Write a complete LC branch and bound algorithm for the job sequencing with deadlines problem. Use the fixed tuple size formulation.
8. Write a LC branch and bound algorithm for the knapsack problem using the fixed tuple size formulation.
9. The postfix representation of an infix arithmetic expression LDR is defined recursively to the postfix representation of L followed by the postfix representation of R followed by 0. L and R are respectively the left and right periods of 0. Consider some examples:

Infix.	Postfix.
(i) a + b	ab +
(ii) (a+B) *C	ab + *
(iii) (a-b)/(c*d)	ab-cd*/

(a) Write an algorithm to evaluate a postfix expression E. Assume E is presented as a string and that there exists an algorithm NEXT-TOKEN(E) that returns the next token (i.e. operator or operand) in E. When all tokens in E have been extracted, NEXT-TOKEN(E) returns. Assume that the only operators in E are binary +, _, * and /. (Hint: Make a left to right scan of E using a stack to store operands and results. When even an operator is run in E, the top two operands on the stack are its right and left operands),

10. Write an algorithm to obtain the postfix form of an infix expression E. Again assume E has only the binary operators +, _, * and /. (Hint: Make a left to right scan of E using a stack to store operators until both the left and right operands of an operator have been output in postfix form). Note that E may contain parenthesis.

Course Outcome (COs)	Description
CO1	Able to design, implement and analysis of standard searching and sorting algorithms.
CO2	Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
CO3	Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
CO4	Able to implement the major graph algorithms and their analysis.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	3	2	2	2				1			1
CO2	2	3	3	2		1			1			
CO3	2	3	3	3								1
CO4	2	2	3	2								

Database Systems
COT-313
(Use Oracle or IBM DB/2)

L	T	P
-	-	3

1. Create a database (database 1) for your class group containing information - roll no., name, group, branch etc. about students.
2. Create a database (database 2) for the employees of an organization and edit it using various available options.
3. Index the database created in problem 3 using various conditions.
4. Use various select queries to search the database for both the databases using different conditions.
5. Generate report for both the d n atabases.
6. Practice various prompt commands like create, select etc.

Application Programs: -

1. Write a program to list out all the information about students getting marks than 70 % using loop structure.
2. Write a program to enter the data into database 2.
3. Write a program to modify the designation of all employees of database 2 for whom the date of joining is before Jan. 10, 2002.
4. Write a program to display 5th record from both the databases.
5. Write a program to delete the records of the students who are getting marks less than 40 %.
6. Develop an MIS for an XYZ Paper Mill to automate its

a. inventory
b. purchase
c. sales
d. payroll
e. annual production report.

Course Outcomes (COs)	Description
CO 1	Learning installation and using a DBMS.
CO 2	Creating small databases using SQL
CO 3	Writing variety of queries for retrieval of data from database
CO 4	Learning to connect database with application

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2		3	1	2	1	2		1	3	2	1
CO 2	1	2	2	3	3		2			3		3
CO 3	1		2	1	3	2			1	2	3	3
CO 4		2		1	2	3		3	2	3	2	2

Software Testing (Pr.)
COT-315

L	T	P
-	-	3

(A) Implement the following in C/C++

1. Develop a formula for the number of robustness test cases for a function of n variables.
2. Develop a formula for the number of robustness worst test cases for a function of n variables.
3. Find a cyclomatic complexity of a graph.
4. Study the development of decision table for the triangle problem.
5. Study the development of decision table for the next date function.
6. Develop a program for the data flow testing.
7. Develop the program for the white box testing.
8. Develop the boundary value analysis (test case) on triangle problem.
9. Develop the boundary value analysis (test case) on next date function.

(B) Developing a small project /tool to generate test data, to execute test data etc.

(C) Exposure to Automated CASE tool

BOOK

1. Paul C. Jorgensen: Software testing--A Craftsman ‘s Approach, II Edition,CROC Press.
2. Pankaj jalote :an integrated approach to software engineering, III Edtion.
3. Art of software testing ,II Edition , john wiley publication, 2004.

Course Outcomes (COs)	Description
CO 1	Insight in importance of testing techniques in software development life cycle
CO 2	Ability to implement structural testing methods efficiently.
CO 3	Ability to implement functional testing.
CO 4	Analysis of the complexity of given program.
CO 5	Knowledge of commonly used testing tools.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	1		2	3	1			1	1	1	2
CO 2	1	1	2	2	3							
CO 3	1	3	1	1	3							
CO 4	1	2	1	1	3							

Operating Systems
COT-302

L	T	P
4	2	

1. File & CPU management

Operating system functions and characteristics, historical evolution of operating systems, Real time systems, Distributed systems, Methodologies for implementation of O/S service system calls, system programs, Interrupt mechanisms, concept of threading.

File Systems: Functions of the system, File access and allocation methods, Directory Systems: Structured Organization, directory and file protection mechanisms, implementation issues; hierarchy of file and device management.

CPU Scheduling: Levels of Scheduling, Comparative study of scheduling algorithms, multiple processor scheduling.

2. Storage & Device Management

Storage Management: Storage allocation methods: Single contiguous allocation, multiple contiguous allocation, Paging, Segmentation combination of Paging and Segmentation, Virtual memory concepts, Demand Paging, Page replacement Algorithms, Thrashing,

Device Management: Hardware Organization, Device scheduling policies & I/O management.

Protection: Mechanisms and Policies, Implementation.

3. Deadlocks & Concurrency Control

Deadlock: Deadlock characterization, Deadlock prevention and avoidance, Deadlock detection and recovery, practical considerations.

Concurrent Processes: Critical section problem, Semaphores, Classical process coordination problems and their solutions, Interprocess Communications, multithreading.

4. Case Studies

DOS: Study of DOS with reference to storage management, device management, file system, interrupt mechanism.

UNIX : Study of UNIX with reference to storage management, file system, concurrency control, CPU scheduling.

BOOKS

1. Peterson, J.L. & Silberschatz, A.: Operating System Concepts, Addison, Wesley-Reading.
2. Brinch, Hansen: Operating System Principles, Prentice Hall of India.
3. Haberman, A.N.: Introduction to Operating System Design Galgotia Publication, New Delhi.
4. Tanenbaum, A.S.: Operating Systems.
5. Hansen, P.B.: Architecture of Concurrent Programs, PHI.
6. Shaw, A.C.: Logic Design of Operating Systems, PHI.

Course Outcomes (COs)	Description
CO 1	Understanding of the inherent mechanism involved in functioning of an operating system.
CO 2	Ability to analyse various scheduling and synchronisation techniques.
CO 3	Knowledge of file systems its implementation and protection.
CO 4	Analysis of memory and device management methodology.
CO 5	Comprehensive outlook in design principles of operating systems.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	3				1					3
CO 2	2	3	3	2		1				1		
CO 3	1		2	1					1			
CO 4	2	3	2	2				1				
CO5	3	2	2	3	3						1	2

Computer Hardware Technologies
COT-304

L	T	P
4	1	-

1. **Memory**
   Memory, memory chips & modules, memory types, advanced memory technologies, troubleshooting memory.

2. **Motherboard**
   PC family tree, motherboard controllers and system resources, input-output ports, IRQ, I/O bus system: ISA, MCA, EISA, VESA local bus, PCI, AGP, PCIX; on board I/O devices, ROMBIOS, ROM POST, CMOS setup.

3. **Power Supply**
   Power supply function and operation, power supply quality and specification, power protection and back-up, backup power system; UPS; troubleshooting power supply.

4. **Interfaces and I/O Ports**
   Floppy disk interface, IDE interface: ATA standards, master-slave configuration, data transfer mode; SCSI interface: SCSI bus, SCSI standards; which is better SCSI or IDE; serial ports, parallel ports, USB, Video adapters, troubleshooting Video adapters.

5. **Device drives and peripherals**
   Floppy disk drive, hard disk drive, CD ROM drive, DVD ROM drive, record able drives, keyboards, mice, printers and monitor, trouble shooting drives and peripherals.

BOOKS

1. Craig Zacker & John Rourtre: PC Hardware- The complete reference, TMH.
2. Mark Minosi: The complete PC Upgrade & Maintenance Guide 4/e, BPB publications.
3. S.K. Chauhan: PC Upgrading, maintenance and troubleshooting guide.

Course Outcome (COs)	Description
CO1	Study all the generation of PC and parts of motherboard, bus system, peripheral devices and storage elements.
CO2	Understand the concepts of power supply and backup system
CO3	Knowledge about the different types of device drives, System interfaces and Input Output ports.
CO4	Assemble all the parts of motherboard related to hardware and troubleshooting each component of PC.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1		2	1	2			1	1		1	2
CO 2			2		2	2	3				2	
CO 3					2					1	2	
CO 4		2	3		3		3		3		3	3

Computer Networks
COT-306

L	T	P
4	1	-

1. **Introduction**
   Network Functions, Network Topology, Network Services, Switching Approaches, Transmission media and systems, multiplexing and signaling techniques, Error detection and correction, Internet checksum algorithm.
2. **Layered Architectures**
   Examples, OSI Reference Model, Overview of TCP/IP architecture, Socket system calls, SNMP, Electronic Mail.
3. **Peer-to-Peer Protocols**
   Protocols, Service Models and End-to-End requirements, ARQ, Sliding Window Protocols, RTP, HDLC, PPP protocols, Statistical Multiplexing.
4. **MAC and LAN Protocols**
   Multiple access communication, Random Access-ALOHA, Slotted-ALOHA, CSMA, CSMA-CD, LAN Standards – Ethernet, Fast Ethernet & Gigabit Ethernet, Bluetooth and WiMax standards.
5. **Packet Switching Networks**
   Packet network topology, Datagrams and Virtual Circuits – Structure of Switch / Router, Connectionless and Virtual Circuit packet Switching, Traffic management and QoS – FIFO, Priority Queues, Fair Queuing, MPLS.
6. **TCP/IP**
   Architecture, Internet protocols – IP packet, Addressing, Subnet addressing, IP routing, CIDR, ARP, RARP, ICMP, Reassembly, IPv6, UDP, Transmission Control Protocol – TCP, DHCP, Mobile IPv6, Internet Routing protocols, Multicast Routing.

BOOKS

1. Leon Garcia and Indra Widjaja: Communication Networks – Fundamental Concepts and Key Architectures, TMH, 2000.
2. A.S. Tanenbaum: Computer Networks, Fourth Edition, Pearson Education, 2003.
3. Forouzan: Data Communications and Networks, Fourth Edition, McGraw Hill, 2007.
4. William Stallings: Data and Computer Communications 5/e, PHI.

Course Outcome (COs)	Description
CO 1	To understand computer network basic, different models used for study of computer networks.
CO 2	To design and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
CO 3	To analyze the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
CO 4	To study the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	1			2	2	1				2
CO 2	3	2	3	1	1	2	1		1	2	1	2
CO 3	2	1	2	2	3	2		2			1	1
CO 4	3	3	3	2	2	2	1	1		2		

Advanced Computer Architecture
COT-308

L	T	P
4	1	-

1. Introduction to Parallel Processing

Parallelism in uni-processor system, pipelining, basic assumptions, design techniques, designing pipelined data path, propagating an instruction queue through the pipeline, pipeline hazards and their detection, forwarding, instruction level parallelism, super-scalar architecture, general pipeline and reservation tables, principles of linear pipelining, Instruction and arithmetic pipeline, principles of designing pipelined processor, micro-programming, general approach, micro-code engine, control store, branching and looping, horizontal and vertical micro-code scheme with example, alternative approach to micro-coding.

2. Issues in the Architecture of Distributed Systems

(a) Introduction, Examples, Distributed Systems versus Parallel Systems, Partial Orders, Models of Distributed Systems, Architectural Models, Fundamental Models, Interleaving Model, Happened Before Model, Potential Causality Model, Appropriate Model, Models Based on states, Deposet.

(b) Program correctness, Correctness criteria, Safety and Liveness properties, Correctness proofs, Predicate logic, Assertional reasoning, Well-Founded sets, Predicate Transformers.

(c) Concept of Logical Time, Logical Clock and its implementation, Limitations of Lamport's Logical Clock, Vector Clock and its implementation, Matrix Clock, and other optimal clocks, Using Induction to prove properties of Clocks, Verifying Clock Algorithms, Events, Process States, Clock Skew and Clock Drift, Synchronizing Physical Clocks, Cristian's Method, The Berkeley Algorithm, The Network Time Protocol, Distributed Debugging.

(d) Synchronization Problems, Mutual exclusion, Shared memory algorithms, Peterson and Lamport's Bakery Algorithm, Token passing algorithms, Suzuki-Kasami algorithm, Raymond's algorithm, Message passing algorithms, Lamport's algorithm, Ricart-Agrawala algorithm, Maekawa algorithm, Leader election, Garci-Molina Bully algorithm, Chang-Roberts unidirectional ring algorithm, Hirschberg-Sinclair bidirectional algorithm, Global States and Consistent Cuts, Chandi-Lamport and Lai-Yang Snapshot Algorithm, Global State Collection, Dijkstra-Scholten and other termination algorithms, Chandi-Misra-Hass and other deadlock detection algorithms.

(e) Fault tolerance, Agreement problems, Asynchronous Distributed Consensus, Impossibility (FLP) result and its proof, Consensus in Synchronous systems, Byzantine Generals Problem (BGP), Consensus with oral and signed messages, Failure detectors.

(f) Naming (Tanenbaum-Steen, Ch. 4), Transactions and Concurrency control (CDK, Ch. 12), Distributed Transaction (CDK, Ch. 13), Replication (CDK, Ch. 14)

BOOKS

1. Hwang and F.A.Briggs: Computer Architecture and Parallel Processing, McGraw Hill.
2. Distributed Systems: Concepts and Design; G Colouris, J Dollimore, T Kindberg 3/e Pearson Ed. 2002. (CDK)
3. Distributed Systems: Principles and Paradigm; Andrew S Tanenbaum, Maarten van Steen 3/e Pearson Ed. 2002.
4. Elements of Distributed Computing, VK Garg, Wiley-Interscience Publishers, 2002.

Course Outcome (COs)	Description
CO 1	Understand fundamental issues in architecture design and their impact on performance
CO 2	Analyse existing architectures and design improved architectures
CO 3	Apply knowledge of uniprocessor systems to introduce parallelism
CO 4	Develop applications to solve computationally intensive problems
CO 5	Design and implementation of high performance computing systems

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2	2	2	3	1	3	3	3	2	2	2
CO 2	1	3	3	3	3	3	2		2	1	2	1
CO 3	3	2	2	3	3	3		1		1	1	1
CO 4	2	3	3	2	2	3	1	2		2	1	
CO 5	1	3	3	3	3	2	1		1	1		1

Operating System (Pr.)
COT-312

L	T	P
-	-	3

1. Study of H/W & S/W requirement of different operating system.
2. Implementation of contiguous, linked and indirect allocation strategies assuming randomly generated free space list
3. Implementation of worst, best & first fit for contiguous allocation assuming randomly generated free space list.
4. Implementation of Compaction for the continually changing memory layout & calculate total movement of data.
5. Calculation of external & Internal fragmentation for different program & for different page size.
6. Implementation of resource allocation graph.
7. Implementation of Banker's algorithm.
8. Conversion of response allocation graph to wait for graph.
9. Implementation of Bernstein's condition for concurrency.
10. Implementation of Fork & Join Construct.
11. Implementation of "Semaphore" for concurrency.
12. Study of system calls and various OS management services in Unix/Linux OS and their implementation.

Course Outcome (CO's)	Description
CO1	Ability to understand resource requirements of different operating systems
CO2	Ability to implement various storage allocation techniques.
CO3	Ability to prevent, avoid, detect and recover from deadlock.
CO4	Ability to implement synchronisation algorithm.
CO5	Ability to use system calls effectively.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**
CO 1	2	1	1		1	1						1
CO 2	3	2	2	1				1				
CO 3	2	3	2	3			1		1	1		
CO 4	3	2	3	3		1	1				1	
CO 5	1	2	1	3	1							

Computer Hardware & Troubleshooting Lab (Pr.)
COT-314

L	T	P
-	-	3

1. To solder and desolder various components.
2. To check and measure various supply voltages of PC.
3. To make comparative study of motherboards; 386, 486, PI, PII, PIII.
4. To observe and study various cables, connections and parts used in computer communication.
5. To study various cards used in a system viz. display card, LAN card etc.
6. To remove, study and replace floppy disk drive.
7. To remove, study and replace hard disk.
8. To remove, study and replace CD ROM drive.
9. To study monitor, its circuitry and various presets and some elementary fault detection.
10. To study printer assembly and elementary fault detection of DMP and laser printers.
11. To observe various cables and connectors used in networking.
12. To study parts of keyboard and mouse.
13. To assemble a PC.
14. Troubleshooting exercises related to various components of computer like monitor, drives, memory, and printers etc.

BOOKS

1. Mark Mines Complete PC upgrade & maintenance guide, BPB publications.
2. Craig Zacker & John Rouske, PC Hardware: The Complete Reference, TMH.
3. Scott Mueller, Upgrading and Repairing PCs, PHI, 1999

Course Outcome (COs)	Description
CO1	To identify, soldering and desoldering of various components
CO2	Knowledge of computer ports, cables, Disk drives
CO3	Understanding of various parts of Input-Output devices.
CO4	Able to assemble a PC and troubleshooting each component

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1		2	1		2			2		1	2
CO 2			2		1				1		1	2
CO 3			2		1				1		1	2
CO 4		2	3			2			3	2		3

Advanced Technologies (Pr.)
COT-316

L	T	P
-	-	3

Exercises

1. Learn Visual Basic environment and develop simple calculator.
2. Generate an editor screen containing menus, dialogue boxes etc. using basic.
3. Create an applet with a text field and three buttons. When you press each button, make some different text appear in the text field. Add a checkbox to applet created, capture the event, and insert different text into the text field.
4. Create an applet with a button and a TextField. Write a handleEvent ( ) so that if the button has the focus, characters typed into it will appear in the TextField.
5. Create your own Java Bean called Valve that contains two properties: a Boolean called "on'' and an integer called "level". Create a manifest file, use **jar** to package your Bean. then load it into the Bean box or into your own Beans-enabled program builder tool e.g. BDK so that you can test it.
6. Develop a servlet that gets invoked when a form on a Web page in HTML is submitted. Create a Cookie object and enter/display value for that cookie.
7. Using VB develop a front end for a contact management program using a flat-file database containing names, addresses, telephone numbers, email addresses etc. You should be able to easily add new names to the database. When typing in the name to be looked up, use automatic name completion.

BOOKS

1. Visual Basic: Complete Reference, TMH, 1999.
2. Java 2: The Complete Reference 4/e: Herb Schildt, TMH, Delhi.
3. JavaBeans Programming from the Ground Up: Joseph O'Neil, TMH, Delhi.
4. Java Servlet: Application development: Karl Moss, TMH, Delhi.

Course Outcome (CO's)	Description
CO1	Implement business problems by writing generic programs (using VB and java).
CO2	Develop server side programs using various J2EE constructs such as servlets and JSPs.
CO3	Update and retrieve the data from the databases using SQL.
CO4	Demonstrate the usage of specific application frameworks to develop console based, GUI based and web based application (using VB and Java).

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1	2	3		1			1	1		1	2
CO 2		3	3	1	2		1		1		1	2
CO 3		3	3	1	2		1		1			2
CO 4		3	3	1	2	1	1		1			2

Digital Signal Processing
COT-340
(Departmental Elective I)

L.	T.	P.
3	2	-

1. Introduction

Signals, Systems and signal processing, classifications of signals, concept of frequency in continuous time and discrete time signals. Analog to digital and digital to analog conversion, Discrete time signals, Discrete time systems, LTI systems, difference equations, implementation of discrete time systems.

2. Z-transform and its Applications

Z Transform, properties of Z transform, Inversion of Z transform, applications of Z transform. Discrete Fourier Transform(DFT), properties of DFT, Linear filtering methods based on the DFT, frequency analysis of signals using the DFT.

3. Fast Fourier transform and its applications

FFT algorithms (Radix 2 FFT) algorithm, Implementation of Discrete time systems, Structures for FIR systems, direct form structure, Cascade form structure, parallel form, structures for IIR systems, cascade, direct form and parallel form structures.

4. Design of Digital Filters

Design of IIR filters, Bilinear transformation and impulse invariance method. Matched Z transformation design of FIR filters with different methods.

BOOKS

1. John G. Proakis and Dimitris G. Manolakis, Digital Signal Processing, PHI
2. Oppenheim & Schaffer, Digital Signal Processing, PHI.
3. Rabiner & Gold, Digital Signal Processing applications.
4. S.K. Mitra, Digital Signal Processing, TMH.
5. S. Salivayhan, A Vallavraj, C. Gnanapriya, Digital Signal Processing, TMH.

Course Outcome COs)	Description
CO 1	Understand and analyze continuous and discrete-time signals to visualize them infrequency and time domain.
CO 2	Implement different transforms to design and analysis DSP systems.
CO 3	Develop program to implement digital filters.
CO 4	Demonstrate DSP techniques in spatial and frequency domains.
CO 5	Ability to apply DSP techniques in both the spatial and frequency (Fourier) domains.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2				3	3	1	3	3		3
CO 2	2	3	2	1	2		3		2	3		3
CO 3	2	2	2	2		2	2		2	3		2
CO 4	3	2	3	2	3	3	1		2	2		3
CO 5	2	3	3	2	3	3	2	1	1	2	1	2

Multimedia Techniques
COT-342
(Departmental Elective-I)

L.	T.	P.
3	2	-

1. **Basics of multimedia technology**
   Computers, Communication and Entertainment; Multimedia -An introduction; Framework for multimedia systems; multimedia devices, CD-Audio, CD-ROM,CD-I; presentation devices and the user interface; multimedia presentation and authoring; professional development tools; LANs & multimedia ;Internet, World Wide Web(World Wide Web) & multimedia ;distribution network-ATM & ADSL; multimedia servers & databases; vector graphics; 3-D graphics programs; animation techniques; shading; anti-aliasing; morphing ;video on demand
2. **Image Compression & Standards**
   Making still images; editing and capturing images; scanning images; computer color models; color palettes; vector drawing; 3-D drawing and rendering; JPEG-objectives and architecture; JPEG-DCT encoding and quantization, JPEG statistical coding; JPEG predictive lossless coding; JPEG performance; Overview of other image file formats as GIF, TIFF, BMP, PNG etc.
3. **Audio & Video**
   Digital representation of sound; time domain sampled representation; method of encoding the analog signals; subband coding; Fourier method; transmission of digital sound; digital audio signal processing; stereophonic & quadraphonic signal processing; editing sampled sound; MPEG Audio; audio compression & decompression; brief survey of speech recognition and generation; audio synthesis; Musical Instrument Digital Interface (MIDI); digital video and image compression; MPEG motion video compression standard; DVI technology; time-based media representation and delivery.
4. **Virtual Reality**
   Applications of multimedia, Intelligent multimedia system, Desktop Virtual Reality (VR), VR operating System, Virtual environment displays and orientation tracking; visually coupled system requirements; intelligent VR software systems.
   Applications of environments in various fields viz. Entertainment, manufacturing, business, education, etc.

BOOKS

1. Villamil & Molina, Multimedia : An Introduction, PHI.
2. Lozano, Multimedia : Sound & Video, PHI.
3. Villamil & Molina, Multimedia : Production, Planning and Delivery, PHI.
4. Sinclair, Multimedia on the PC, BPB.
5. Tay Vaughan, Multimedia :Making it work, TMH

Course Outcome (COs)	Description
CO1	Gain insight of basic requirements and need of multimedia communication, multimedia over network, and general multimedia techniques.
CO2	Learn several multimedia data in various digital format, image, audio and video compression techniques and its representation.
CO3	Ability to design and implementation of image, audio and video in virtual environment i.e., entertainment, manufacturing, business etc.

Course Outcomes COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	1								1		1
CO 2	2	3	2	2	2	2	2		1		1	
CO 3	2	3	3	2	2	2	2			2		2

Graph Theory and Combinatorics

COT-344
(Departmental Elective I)

L	T	P
3	2	-

1. Introduction

Basic concepts, subgraphs, vertex, degrees, walks, paths, circuits, cycles, trees, spanning trees, cut vertices and cut edges, connectivity, Euler tours and Hamiltonian cycles, matching perfect matching, connectivity and separability, network flows, 1-isomorphism and 2-isomorphism.

2. Advanced Features

Vertex coloring, chromatic polynomial, edge coloring, planar and non-planar graphs, Euler's formula, Kuratowski's theorem, test for planarity, directed graphs, tournaments, networks, max flow, min cut theorems, graph enumeration, Polya's counting theorem

3. Graph Algorithms

Computer representation of graph, shortest path algorithms, minimal spanning tree, fundamental circuit, depth first search, planarity testing, directed circuits, isomorphism, performance of graph theoretic algorithms.

4. Combinatorics

Basic combinatorial numbers, recurrence relations, generating functions, multinomial, counting principles, Polya's theorem, inclusion and exclusion principles, block design and error correcting codes, Hadamard matrices, finite geometry.

BOOKS

1. Deo N.: Graph Theory and Applications, Prentice Hall
2. D.B.West: Introduction to Graph Theory, Prentice Hall
3. S.A.Choudum: A First Course in Graph Theory, MacMillan [India]
4. V.Krishnamurthy: Combinatorics--Theory and Applications, Affiliated East-West
5. Alan Tucker: Applied combinatorics, Wiley.

Course Outcome (COs)	Description
CO 1	Understand the basic concept of graph, trees and cycles
CO 2	Analyze NP problems to solve various coloring problem
CO 3	Implement various graph traversal algorithms
CO 4	Demonstrate knowledge of the basic concepts of combinatorics

Course Outcomes COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	1	2	2	3	1		1	2	1	
CO 2	1	3	1	2	1		1		1	2	1	1
CO 3	3	2		2	2	1	2	2	1	1	2	1
CO 4	2	2	2	1	2	2	2	1	1	2	1	1

Advanced Database Systems
COT-346
(Departmental Elective I)

L	T	P
3	2	-

1. **Parallel & Distributed Data bases**
   Architecture for parallel databases, Parallel query evaluation, parallel zing individual operations, parallel query optimization; Introduction to distributed databases, distributed DBMS architectures, storing data in a distributed DBMS, distributed catalog management, distributed query processing, updating distributed data, introduction to distributed transactions, distributed concurrency control, recovery.
2. **Data Mining**
   Introduction, counting co-occurrences, mining for rules, tree structured rules, clustering, similarity search over sequences.
3. **Object Database Systems**
   User defined ADT, structured types, objects & reference types, inheritance, design for an ORDBMS, challenges in implementing an ORDBMS, OODBMS, comparison of RDBMS with OODBMS & ORDBMS.
4. **Advanced Topics**
   Advanced transaction processing, integrated access to multiple data source, mobile databases main memory databases, multimedia databases, GIS, temporal & sequence databases.

BOOKS

1. R. Ramakrishnan & J. Gehrks Database Management Systems MGH, International Ed., 2000.
2. Korth, Silberschatz, Sudershan: Data Base concepts, MGH, 2001.
3. C. J. Date, Database Systems, 7th Ed., Addison Wesley, Pearson Education, 2000.

Course Outcome (CO's)	Description
CO 1	To familiarize students with parallel and distributed databases: need and related issues for the design and development
CO 2	To familiarize students with data mining system and process
CO 3	To familiarize students with object oriented concepts and OODBSs.
CO 4	To familiarize students with advanced topics in the area of database technology

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2	3	1		1						
CO 2	3	3		3	2							
CO 3	3	3	3	2	2							
CO 4	3	3	3	2								

Logic of Programming

COT-348

(Departmental Elective I)

L	T	P
3	2	-

1. **Fundamentals**
   Propositions, Tautologies, Precedence rules, System definition, Reasoning using Transformations, Formal Systems, Axioms, Inference Rules, Predicates, Quantification, Free and bound identifiers, Data Values & Types, Generators, semantic definitions of functions, Generator Induction, definedness condition.
2. **Semantics**
   Predicate Transformers, various commands, Alternative and Iterative commands, Procedure call, The characterization of semantics, The semantic characterization of programming language, Two Theorems, Design of Properly terminating constructs, Euclid's Algorithms, Interrupts, spin locks,
3. **Communicating Sequential Processes (CSP)**
   Parallel commands, Coroutines, Subroutines and data representation, Monitors and scheduling, Integer semaphore, Dining Philosophers Problem.

BOOKS

1. David Gries, The Science of Programming, Narosa Publishing House
2. E.W. Dijkstra, A Discipline of Programming PHI
3. Hoare and Jones, Essays in Computing Science, TMH.

Course Outcome (COs)	Description
CO1	Able to understand the fundamentals of logic of programming including formal systems, inference rules and quantification
CO2	Detail study of semantics including predicate transformers, Euclid theorems.
CO3	Be familiar with the concepts of interrupts and spin locks
CO4	Develop an understanding of the communicating sequential processes such as parallel commands, integer semaphore and dining philosophers problem

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3	3	2	1							2
CO 2	2	2	3	3	2							2
CO 3	2	3	3	2	2							1
CO 4	2	2	2	2	2							2

Computer Graphics
COT-350
(Departmental Elective I)

L T P

3 2 -

1 Basic Concept

Introduction, Point Plotting technique, Coordinate System, Line drawing algorithm, Circle generators, Line drawing display, storage tube display, refresh line drawing display.

2 Computer Graphics Devices

Point and positioning drive, light pen, mouse, tablet, Input technique, Positioning technique, and character recognition.

3. 2-D and 3-D transformation

Basic transformation, homogenous coordinate system, composite and other transformation, Point and line clipping, polygon clipping, text clipping, view planes.

6. Interactive Raster Graphics

Raster graphics fundamental, solid area Scan Conversion, Interactive raster graphics, Raster graphics systems.

7. 3-D Graphics

Curve and Surfaces, Bezier and B-spline method, perspective depth, Hidden Surface elimination, depth buffer algorithm, scan line coherence and area coherence algorithm, priority algorithm.

8. Graphics Systems

Device Independent graphs system, Graphics System design, Case Study of Graphics, Kernel System.

BOOKS

1. Hearn and Baker: Computer Graphics: 2nd Ed., PHI.
2. Rogers: Principles of Computer Graphics, MGH
3. Folay: Fundamentals of Interactive Computer Graphics, Addison Wesley.
4. Harrington: Computer Graphics – A Programming approach.
5. Newmann and Sproull: Introduction to Interactive Computer Graphics.

Course Outcome (COs)	Description
CO1	To learn algorithms to build the basic blocks of graphic systems.
CO2	To learn functionalities of input and output devices for graphic systems
CO3	To learn modelling, manipulation, clipping and rendering operations for graphic systems.
CO4	To learn shapes of objects in various curves and splines methods.

Course Outcomes (COs)	Programme Outcomes(POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	1	2		1	1			1			1
CO2	1											1
CO3	2	1	2	1	2			1	1			1
CO4	2	1		2					1			1

Software Quality and Reliability
COT-352

L	T	P
3	2	-

1. **Software Quality**
   Meaning and scope , software quality factors, software quality metrics, relationship between quality factor and quality metrics, quality management systems, software reviews, formal technical review, correctness proof, statistical quality assurance, clean room software engineering, ISO 9001 and SEI-CMM standards of software quality assurance
2. **Software Reliability**
   Meaning and its relation with software quality , reliability modeling-exponential failure time models (viz. Jelinski Morana model, Schneidiwind's model, Musa's basic executionj time model, hyperexponential model), Weibull and Gamma failure time models (viz. Weibull model and S-shaped reliability growth model) , and infinite failure category models (viz. Duane's model, geometric model, Musa – Okumto model). 3 **Software Testing**
   Meaning, scope and its relationship with software quality: software testing testing techniques: white box testing, basis path testing, control structure testing, and black box testing etc. ; software testing strategies – unit testing, integration testing, validation testing and system testing etc.

BOOKS

1. Software Quality: Concepts and Plan, By Robert H Dunn, Prentice Halll International.
2. Software Reliability: Measurement, Prediction and application, By Lohn D Musa, McGraw Hill
3. Software Reliability Engineering By Michele R Lyu, McGraw Hill.
4. Effective Methods of Software Testing, By William E Perry, Wiley.
5. Concepts of Reliability, By L Srinath.
6. Software Reliability, By K K Aggarwal.
7. Software Reliability, By H Koptez.

Course Outcome (COs)	Description
CO 1	Apply ISO/SEI Standards for Correctness proof, Software Quality Management and Software Quality Assurance.
CO 2	Apply Reliability modeling techniques for different failure time models and understanding its relation with Software Quality.
CO 3	Understand different testing strategies, apply according at each level to makes sure of the Customer's reliability and their satisfaction in the application.

Course Outcomes (CO's)	Programme Outcomes(PO's)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2	2	2	3	-	2	-	1	1	3	1
CO 2	3	3	3	3	3	-	2	1	2	2	3	
CO 3	3	3	2	3	3	3	3	2	3	3	3	1

Internet and Intranet Engineering
COT-401

L	T	P
4	1	-

1. **Introduction**
   Introduction to Internet and Intranet, Review of TCP/IP model, UDP, IPv6 Protocol, IP/IPv6 addressing, IP datagram, IP/IPv6 addressing, IP datagram Routing and Upper Layer Protocols: CIDR, Routing, Multicasting, Mobile routing, SMTP, FTP, HTTP, Telnet Protocols. Superserver xinetd, ARP Tables, How DNS Works
2. **Intranet Technologies**
   Internet Technologies integrated with LAN technologies, Web-Server, E-Mail, Printing and file Services, FTP, Net-News, Network Operating Systems: Introduction to Linux, Netware, Windows NT/2000 service models. Centralized Services Model.
3. **Authentication, File and other Directory Services**
   File Servers: SMB Protocol, NFS, NIS+, Distributed File Services (WinNT), Unix/Linux File Permissions Model, The WindowsNT domain+user model of authentication, The Linux Private Groups for users on the system. LDAP, Kerberos, rsh, rcp, rlogin, Microsoft Active Directory Service, Novell Directory Service, Pluggable Authentication Module (PAM), auth_ldap.
4. **Web Services**
   The Apache Web Server, Virtual Hosts, Application Support: CGI, Fast-CGI, mod_perl, PHP4, Java Servlets Support through Tomcat as DSO module of Apache, IIS web server, ISAPI, NSAPI, Frontpage Extensions, Frontpage extensions support for Apache.
5. **Security and Control**
   TCP wrappers, SSH, Firewall, IP masquerading, IP chains, Advanced routing techniques, Network Address Translation, Proxy Servers: Hardware and Software.

BOOKS

1. Daniel Minoli, Internet & Intranet Engg. (Tech., protocol & application), MGH.
2. Red Hat Linux Bible, IDG.
3. Windows 2000 Complete Reference TMH.
4. Netware 5 Complete Reference, TMH.
5. Daniel J. Barrett and Richard Silverman, SSH, The Secure Shell: The Definitive Guide, O'Reilly.
6. Charles Aulds, Linux Apache WebServer Administration, Sybex.

Course Outcome (COs)	Description
CO1	To understand the advanced concepts of addressing and routing in Internet technology
CO2	To analyze various components, related techniques, and service models of Internet Technology
CO3	To study the file servers, file permission models, authentication models, and directory services
CO4	Able to understand various techniques to design and develop web services.
CO5	To apply security aspects in providing Internet based services

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	2			2					1		
CO2	2		3									
CO3	1	2										
CO4	3			3	3				2		2	
CO5	2	3	2		2	2		1				1

Microprocessors - II
COT-403

L	T	P
3	1	-

1. **Pentium Architecture**
   Basic architecture of P II/ P III, Memory management models, registers and flags.
2. **Instruction**
   Basic data types in P II/ P III, addressing modes, instruction format and types, instruction set and prefixes.
3. **Memory Management**
   Modes of operation of P II/ P III – real, protected and virtual; segmentation and its working, paging and its working, concept of descriptor table, superscalar architecture.
4. **Advanced Concepts**
   Branch prediction, exception handling, performance measurement, optimization, MMX – register and instruction set, interrupt handling.

BOOKS

1. Bray, Intel Microprocessors, The 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium & Pentium Processor - Architecture, Programming and Interfacing, PHI.
2. A.K. Ray & K.M. Bhurchandi, Advanced Microprocessors & peripherals-Architecture, Programming & Interfacing: TMH.
3. Intel, Pentium Processor Data Handbook, Intel, 1999.

Course Outcome (COs)	Description
CO 1	Understand the basic architecture of P-II/P-III microprocessor and different memory management models available.
CO 2	Understand the instruction set of P-II/P-III microprocessor.
CO 3	Understand the implementation of memory management concepts in P-II/P-III.
CO 4	Understand and analyze advance concepts like branch prediction and branch handling etc.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO 12**
CO 1	3	3	2	3	2	1			1			1
CO 2		1	3		2				1	1		1
CO 3	3		2	3	3				1	1		
CO 4	1	3		3		1						1

Statistical Models for Computer Science
COT-405

L	T	P
3	1	-

1. **Random Variables**
   Introduction , Engineering Applications to Random variables, Discrete random vectors, Probability density function (pdf), Cumulative Distribution functions (cdf), Probability Generating Function, Continuous random variables: some continuous distributions (Exponential, Hyperexponential, Erlang, Gamma, Normal), Functions of random variables, jointly distributed random variables.
2. **Expectation**
   Introduction, Moments, Expectation of functions of more than one random variable, Brief introduction to Conditional pmf, pdf and expectation, Moments and transforms of some distributions (Uniform, Bernoulli, Binomial, Geometric, Poisson, Exponential, Gamma, Normal), Computation of mean time to failure.
3. **Stochastic Processes**
   Classification of stochastic processes, The Bernoulli process, The Poisson process, renewal process, renewal model of program behaviour.
4. **Markov Chains**
   Computation of n-step transition probabilities, State classification and limiting distributions, Distribution of times between state changes, Irreducible finite chains with aperiodic states, M/G/1 queuing system, Discrete parameter Birth-Death processes, Analysis of program execution time. Continuous parameter Markov Chains, Birth-Death process with special cases, Non-Birth-Death Processes.
5. **Estimation and Regression**
   Maximum likelihood estimation (MLE), Bayesian Estimation Techniques,
   Expectation Maximization, Linear Regression, Application of Regression in pattern recognition

BOOKS

1. K.s. Trivedi, Probability, Statistics with reliability, Queuing and Computer Science Applications, PHI, 2001.
2. The Probability Tutoring Book (An intuitive Course for Engineers & scientists) by Corol Ash IEEE Press.
3. A first course in Probability by Sheldom M. Ross Rs 350/-
4. Introduction to Probability Models by Sheldom M. Ross Rs 450/-
5. Statistics foe the Engineering and Computer Sciences by William Mendenhall.

Course Outcome (CO's)	Description
CO 1	Understanding to the basic concepts of Probability and Bayes' Estimation.
CO 2	Apply Probability theory to formulate discrete and continuous distributions
CO 3	Acquire different optimization functions and methods
CO 4	Develop and classify the stochastic model for real-world optimization
CO 5	Apply Markov chain, queuing theory and Birth Death process in computation algorithms

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1	2		2	2	1	2	2	1	1	2	
CO 2	2	2	2	1	2	2	2	1	1	2	1	
CO 3	3	2		2	1	2	2	1	1	2	3	
CO 4	2	1	1	2	2	3	1		1	2	1	
CO 5	1	3	1	2	1		1		1	2	1	

Computer Networks (Pr.)
COT - 411

L	T	P
-	-	2

1. Define Ethernet, Token Ring, Token Bus, Aloha in user manual 1 of component 1.
2. Create a hub connected LAN with four nodes in your lab and using traffic capture software measure throughout. Compare with simulated LAN of similar configuration and determine for what value of p-Persistent throughput remains unchanged in user manual 1 of component 2.
3. Define TCP/IP in user manual 1 of component 8.
4. Define Router in user manual 1 of component 5.
5. Simulate and Ethernet LAN using 6 to 15 nodes, apply p-Persistent (1/6 to 1/15) and compare the performance in terms of throughout, probability of success and average attempts. Plot the following
6. Average attempts Vs Normalized throughout
7. No of transmitting nodes Vs Probability of success
   a. No of transmitting nodes Vs Normalized throughput
   b. Refer page 257,263 and 281 of Computer Networks by Andrew S.
   c. Tanenbaum (4^{th} Edition) in user manual 2 of component 1.
8. Simulate an ALOHA and Ethernet network for 6 to 15 nodes and compare performance in user manual 2 of component 1.
9. Simulate an Ethernet LAN using 6 to 15 nodes, change error rate (10^{-6} to 10^{-9}) and compare with throughput, end time, delay and response time in user manual 2of component 1.
10. Setup/Stimulate and Ethernet network of 6 to 15 nodes, for 1-Persistent, vary number of transmitting nodes and compare collision count, throughput, end time, delay and response time in user manual 2of component 1.
11. Stimulate a wireless network of 15 node and compare performance with Ethernet and Token ring in user manual 2 of component 1.
12. Stimulate a wireless network of 15 node and compare performance with Ethernet and Token ring in user manual 2 of component 3.
13. Compare working and performance difference between simple BSS and multiple BSS with transmitting node in wireless LAN by simulation or real time in user manual 2 of component 3.
14. Compare the performance of store and forward, cut through and fragment three switches in Ethernet protocol in user manual 2 of component 4.
15. Simulate a WAN network using TCP/IP protocol. Keep the error rate moderate (from 10^{-7} to 10^{-9}). Change the congestion algorithm then compare the network performance in user manual 2 of component 8.
16. Simulate an WAN network using Router. Change the routing protocol, and then compare the simulation time in user manual 2 of component 9.
17. Simulate an WAN network using Router. Change the scheduling technique, and then compare the Queuing delay for different prioritized packet in user manual 2 of component 10.

Course Outcome (COs)	Description
CO1	To be able to understand various components and protocols of Computer Networks.
CO2	Ability to analyze and simulate different protocols.
CO3	To be able to simulate, analyze, and evaluate performance of Ethernet LAN under different implementation conditions.
CO4	To be able to simulate, analyze, and evaluate performance of WAN under different implementation conditions.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1		2		1								2
CO2	2	2	1	2	3				2			1
CO3	3	2	3	3	3		1		2	1		
CO4	3	2	3	3	3		1		2	1		

Microprocessors - II (Pr.)
COT-413

L	T	P
-	-	2

Note: Using 32-bit processor and assembler develop programs.

1. Write a program to read a 32-bit integer, store that number in EAX register and display it digit by digit.
2. Repeat exercise 2 for 64 bit number using two 32 bit registers.
3. Write a program to find factorial of a number, where result does not exceed 64 bit. Use procedure to calculate factorial and pass parameters.
4. Write modular program to perform addition, subtraction, multiplication and division of two 32-bit numbers.
5. Sort n 32-bit numbers using modular program.
6. Write a program that generates a large number of data cache misses and compare its results with a program having large number of hits.
7. Repeat exercise 6 for code cache.

Course Outcome (COs)	Description
CO1	To understand the generalized architecture of advanced microprocessors.
CO2	Identify, formulate and solve engineering problems in microprocessor based system design.
CO3	To understand Modular Programming for 32-bit assembler
CO4	Use design tools for microprocessor system design, test and evaluation.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**O9**	**PO10**	**PO11**	**PO12**
CO1	2	1		2	1	1	1	1	1			
CO2	1	2	3	1	2				1			1
CO3	2	3	3	2	1	1		2	1	1	2	1
CO4	2	2	2	3	2				1	1	1	

Minor Project
COT-415

Course Outcome (COs)	Description
CO1	Ability to apply knowledge of mathematics, science &engineering in practice and identify, critically analyze, formulate& solve engineering problems with select appropriate engineering tools and techniques and use them with dexterity.
CO2	Demonstrate and able to communicate effectively, appreciate the importance of goal setting and to recognize the need for life-long learning.
CO3	Design a system & process to meet desired needs within realistic constraints such as health, safety, security and manufacturability with devise & conduct experiments, interpret data &provide well informed conclusions.
CO4	Acquire to system integration skills, documentation skills, project management skills and problem solving skills.

Course Outcomes (Cos)	Programme Outcomes (Pos)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**O9**	**PO10**	**PO11**	**PO12**
CO1	2	1		2	1	1	1	1	1			
CO2	1	2	3	1	2				1			1
CO3	2	3	3	2	1	1		2	1	1	2	1
CO4	2	2	2	3	2				1	1	1	

Seminar
COT-417

Course Outcome (COs)	Description
CO1	Able to communicate and debate about a given topic
CO2	Able to arrange, organize and present the conceptual and technical ideas in systematic way
CO3	Able to write technical documents, presentation and give the oral presentation of a given topic
CO4	Able to analyze the functioning of different approaches of a topic and should have fundamental knowledge of discussed topic
CO5	Able to explain the concept of different approaches of a problem

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**O9**	**PO10**	**PO11**	**PO12**
CO1		1		1	1	2	1		2	3	2	1
CO2	1	3	2	3	2	2	2	2	3	3	2	2
CO3	1	1	2	2	1	2	2	2	2	3	3	2
CO4	2	3	2	2	3				2	1	3	2
CO5	2	1	2	1	1	1	1	1	2	3	3	1

Software Project Management
COT-441

L	T	P
3	1	-

1. **Conventional Software Management**.
   Evolution of software economics. Improving software economics: reducing product size, software processes, team effectiveness, automation through software environments. Principles of modern software management.
2. **Software Management Process**
   Framework,: Life cycle phases- inception, elaboration, construction and training phase. Artifacts of the process- the artifact sets, management artifacts, engineering artifacts, pragmatics artifacts. Model based software architectures. Workflows of the process. Checkpoints of the process.
3. **Software Management Disciplines**
   Iterative process planning. Project organisations and responsibilities. Process automation. Project control And process instrumentation- core metrics, management indicators, life cycle expections. Process discriminants.

BOOKS

1. Software Project management, Walker Royce, Addison Wesley, 1998.
2. Project management 2/e , Maylor.
3. Managing the Software Process, Humphrey.
4. Managing global software Projects, Ramesh, TMH,2001.

Course Outcome (COs	Description
CO 1	Apply models based empirically derived formulas to find scope of project and to estimate the project cost based on LOC or FPs by using.
CO 2	Identify internal attributes (Maintainability, Flexibility, Portability, Re-usability, Readability, Testability, and Understandability.)Apply internal attributes to evaluate External Quality Characteristics: Correctness, Usability, Efficiency, Reliability, Integrity, Adaptability, Accuracy, and Robustness.).
CO 3	Understand different types of relationship exist among data and control structures of different modules of a project or within same module of a project.
CO 4	Select McCabe's Cyclomatic Complexity, Cohesion, Coupling and Function Points for analyzing internal quality.

Course Outcomes (COs)	Programme Outcomes (POs)										
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
CO 1	3	2	3	2	1	-	1	3	1	2	3
CO 2	3	3	3	2	3	3	2	2	3	3	3
CO 3	2	3	3	3	3	1	2	1	2	3	3
CO 4	3	3	3	2	3	-	2	1	1	3	3

Fuzzy Logic
COT-443
(Departmental Elective II)

L	T	P
3	1	-

1. **Introduction**
   Background, uncertainty and impression, Statistics and Random Processes, Uncertainty in Information, Fuzzy sets and Membership, Chance versus Ambiguity, Classical Sets – Operations, Properties, mapping to classical sets to functions; Fuzzy Sets – Operations and Properties; Sets as points in Hypercubes.
2. **Relations and Functions**
   Cartesian Product, Crisp relations – cardinality operations, properties, composition, Fuzzy Relations – Cardinality operations, properties, Fuzzy Cartesian Product and Composition, Noninteractive Fuzzy Sets, Tolerance and Equivalence Relations, Crisp Equivalence Relation, Crisp Tolerance Relation, Fuzzy Tolerance and Equivalence Relations, Value Assignments, Cosine amplitude, Max-Min method, other similarity methods, Membership Functions – Features, Standard forms and biyearlies, Fuzzyfication, Membership value assignments, Intuitions, Inference, Rank Ordering, Angular Fuzzy sets, Neural Networks, Genetic Algorithm, Inductive Reasoning. Lambda-Cuts for Fuzzy Sets, Lambda-cuts for fuzzy relations, Defuzzification Methods.
3. **Arithmetic and Logic**
   Extension Principle, Crisp functions, Mapping and Relations, Functions of Fuzzy Sets, Fuzzy Transform Practical Considerations, Fuzzy Numbers, Interval Analysis in Arithmetic, Approximate Methods of extension, Vertex Method, DSW Algorithm, Restricted DSW Algorithms, Comparisons, Fuzzy Vectors, Classical predicate logic, Tautologies, Contradictions, Equivalence, Exclusive Oral Exclusive Logical proofs, Deductive Proofs, Deductive Inferences, Fuzzy Logic, Approximate Reasoning, Fuzzy Tautologies, Contradictions, Equivalence and Logical Proofs, other forms of the implication operation, other forms of the composition operation.

BOOKS

1. Timothy J Ross, Fuzzy Logic with Engineering Applications, MGH.
2. Klir and Yuan, Fuzzy Sets & Fuzzy Logic-Theory and Applications, PHI.
3. Klir & Folger, Fuzzy Sets, Uncertainty and Information, PHI.

Course Outcome (COs)	Description
CO1	Understand the fundamentals of crisp set and fuzzy set
CO2	Apply knowledge of fuzzy set related notions, relations and functions to engineering problems
CO3	Comprehend neuro-fuzzy principles and their computational compatibility
CO4	Demonstrate and apply arithmetic and logic functions to implement fuzzy algorithms

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	2	2								1
CO 2	2	3	3	3	1							1
CO 3	2	2	2	2	1							
CO 4	2	3	3	2	1							1

Parallel Computing
COT-445
(Departmental Elective II)

L	T	P
3	1	-

1. **Introduction**
   Parallel processing terminology, Pipelining Vs Data parallelism, Control parallelism, Scalability, Control parallel approach, Data parallel approach, Data parallel approach with I/O
2. **PRAM Algorithm**
   Parallel reduction ,Prefix sums, List ranking, Preorder tree traversal, Merging two sorted lists, Graph coloring, Reducing the number of processors, Problems defying fast solutions on PRAMS
3. **Parallel Programming Languages**
   Programming parallel processes, Example and application, C* programmers model, Language features, Sample program, OCCAM, programmer's model, Language constructs, Sample program, C-LINDA, Programmers model, Language constructs, Sample program
4. **Mapping and Scheduling**
   Mapping data to processors on processor arrays and multicomputers, Dynamic Load Balancing on multicomputers, Static scheduling on UMA multiprocessors, Deadlock.
5. **Elementary Parallel Algorithms**
   Classifying MIMD algorithms, Reduction, Hypercube SIMD model, Shuffle-Exchange SIMD model, 2-D Mesh SIMD model, UMA Multiprocessor model, Broadcast, Prefix sums
6. **Matrix Multiplication**
   Sequential matrix multiplication, Algorithms for processor array, Algorithms for multiprocessors, Algorithms for multicomputers
7. **Sorting**
   Enumeration sort, lower bound on parallel sorting, Odd-even transposition sort. Bitonic merge, Quick sort based algorithms, Random read and random write.

BOOKS

1. Michael Quinn: Parallel Computing-Theory and Practice, MGH.
2. Ed. Afonso Ferreira and Jose' D. P. Rolin, Parallel Algorithms for irregular problems - State of the art, Kluwer Academic Publishers.
3. Selim G. Akl, The Design and Analysis of Parallel Algorithms, PH International.
4. Brassard and Bratley, Fundamentals of Algorithms, PHI, New Delhi

Course Outcome (CO's)	Description
CO 1	Providing an exposure to current and emerging trends in parallelism concept and analyze fundamental issues of data parallel approach with its impact on performance.
CO 2	Design and implementation of high performance computing algorithms.
CO 3	Enhance the programming skills with different parallel computing programming languages.
CO 4	Understand the principles of mapping data to multiprocessor and implement scheduling algorithms
CO 5	Design and implementation of various matrix multiplication and sorting algorithms.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2	2	2	3	1	3	3	3	2	2	
CO 2	1	3	3	3	3	3	2		2	1	2	1
CO 3	3	2	2	3	3	3		1		1	1	2
CO 4	2	3	3	2	2	3	1	2		2	1	
CO 5	1	3	3	3	3	2	1		1	1		2

Image Processing
COT – 447
(Departmental Elective II)

L	T	P
3	1	-

1. Introduction

Image Processing Fourier Transform and Z-Transform, Causality and stability, Toeplit and Circulate Matrics, orthogonal and unitary Matrices and Kroenker product, Markov Processes KI Transform Mean square Estimates and Orthogonal Principles.

Image sampling quantization, Band Limited Image Sampling Versus Replication, Reconstruction of Image from samples Sampling Theorem, Sampling Theorem for Random Fields, Sampling Optimal Sampling, Nonrectangular Grid Sampling, Sampling Aperture, Display Aperture/ Interpolation Functions, Lang range Interpolation, Moire Effect. Image Quantization Uniform Optimal Quantizer, Properties of Mean Square Quantizer, Commander Design Visual Quantization

2. Image Transforms

Two Dimensional Orthogonal and Unitary Transforms and their properties. One Dimensional and Two Dimensional DFT Cosine and Sine Transforms Iiadmard, Slant, IIARR and KI, Transforms and their properties, Approximation to KI Transforms. Image representation by stochastic models, One Dimensional Causal Models, AR and ARMA models, Non Causal Representation Spectral factorization, Image Decomposition.

3. Image Enhancement and Restoration

Point Operation, Histogram Modeling, Spatial Operations, Transform Operations, Multispectral Image Enhancement. Image Observation Models, Inverse and Wiener filtering; FIR Wiener Filters, Filtering using Image Transform Causal Models and recursive filtering Maximum entropy restoration. Extrapolation of band limited signal.

4. Image Analysis and Image Compression

Spatial feature extraction, Edge detection and boundary extraction Boundary, region and moment representations structures, Texture, Image Segmentation, Reconstruction from Projections, Pixel Coding, Productive Techniques, Transform Coding Theory, Coding of Image, Coding of two-tone image.

BOOKS

1. Anil Jain: Digital Image Processing
2. Gonzalez Woods: Image Processing

Course Outcome (CO's)	Description
CO 1	Acquire the fundamental concepts of a digital image processing system
CO 2	Apply knowledge of various mathematical tools used for 1D and 2D signal analysis and processing
CO 3	Analyze 2D signals in the spatial and frequency domain
CO 4	Design and implement algorithms for digital image processing operations
CO 5	Ability to apply image processing techniques to solve various real time problems

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	2	1			1				1		1
CO 2	2	2	3	2	2		1	1	2	1		1
CO 3	3	3	2	2	2				1			1
CO 4	2	2	3	2	3	1			2	2		1
CO 5	3	3	3	2	2	2	1	1	1	1		1

VLSI Technology
COT-449
(Departmental Elective II)

L T
3 1

1. Crystal Growth
MGS,EGS, Czochralspi crystal Puller, Silicon shaping, Wafer Preparation. Epitaxy: Vapour Phase Epitaxy, Epitaxial Layer evaluation, Molecular Beam Epitaxy.

2. Oxidation
Thermal Oxidation Kinetics, Oxidation Techniques, Oxide Properties, Oxidation induced Defects. Lithography: Photolithography, E-beam lithography, X-ray Lithography.

3. Reactive Plasma Etching
Plasma Properties, Feature Size control and anisotropic etching, Plasma etching techniques and equipment. Di-electric and Poly-Silicon Film Deposition: Deposition Processes for Poly- Si, SiO2, SiO3N4; Plasma assisted Depositions.

4. Diffusion
A Qualitative view of atomic diffusion in Solids, diffusion mechanisms, Fick's one dimensional diffusion equation, constant source and limited source diffusion, Diffusion of Grp3 and 5 impurities in Silicon Impurity Sources, diffusion apparatus, Characterization of diffused layers.
Ion Implantation: Introduction, Range Theory, Implantation Equipment Annealing.

5. Metallization
Metallization applications, Choices, Physical Vapour Deposition. Sputtering, Metallization Problems. Assembly & Packaging : Package Types, design considerations, Package fabrication technologies, Future trends.

6. Isolation Techniques
Bipolar IC fabrication Process Sequence. n MOS IC fabrication Process Sequence.

BOOKS

1. S.M. Sze VLSI Technology, Mc Graw Hill (1988)
2. S.K. Ghandhi VLSI Fabrication Principles.

Course Outcome (COs)	Description
CO1	Study about the Natural and Artificial growth of Crystals, Oxidation and Lithography techniques.
CO2	Knowledge of the Fabrication of integrated circuits of Plasma and Film Deposition.
CO3	Ability to understand Atomic Diffusion, Metallurgy and Isolation process.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1											
CO2		1	2			1			1		1	
CO3	1	1	1			1						

Security and Cryptography
COT-479

L	T
3	1

1. Traditional Cryptography

Cryptoanalysis, substitution and transposition ciphers, Cryptographic principles, secret-key algorithms: DES, DES chaining, Breaking DES, IDEA, Differential and Linear cryptoanalysis Public-key algorithms: RSA, Knapsack

2. Authentication protocols

KDC protocol, shared secret key, Diffle-Hellman key exchange, Needham-Schroeder protocol, using Kerberos, interlock protocol, digital signatures – secret key and public key signatures, DSS, message digest, MD5 and secure hash algorithms

3. Computer Security Mechanisms

Role of different security mechanisms, passwords – technology and administration, principles of database system security, epidemic of viruses: types of viruses, study of different virus codes, means of spread, prevention from virus, life cycle of a virus, immunization, Trojan horse and bombs with examples, writing antivirus/trojan codes.

4. Network Security

Basics, security functions, preventing loss and damage, securing local area network – authorization, security plan and policy, securing enterprise network – setting priorities, security plans, securing network components, hardware security, levels of access control and authorization.

BOOKS

1. Richard H. Baker, Network Security, McGraw Hill International Ed.1996
2. B. Schneier, Applied Cryptography, John Wiley New York, 1996.
3. C. Kaufman et. al, Network Security, Prentice Hall International, 1998.

Course Outcome (COs)	Description
CO1	Able to design and implement secure cryptosystem - symmetric as well as asymmetric, and authentication protocols.
CO2	Able to understand and choose suitable cryptosystem to secure existing real time application.
CO3	Able to analyze the strength of existing cryptosystem - symmetric and Asymmetric, and authentication protocols, in terms of security.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	3	1	2	3	2	2	2	3	2	1
CO2	2	3	1	2	1	2	3	1		1		2
CO3	3	3	2	3			1	2		1		2

Fundamentals of Software Engineering
COT-471
(Open Elective I – For branches except CO, IT)

L	T	P
3	1	-

1. **Introduction**
   Introduction to S/W crisis & S/W processes; S/W life cycle models Build & Fix, waterfall prototype evolutionary, spiral model.
2. **Requirement Analysis & Specifications**
   Problem Analysis – DFD, Data dictionaries, ER diagrams, object diagrams; approaches to problems analysis; SRS; specifying behavioral & non behavioral requirements.
3. **S/W Design**
   What is design? Modularity, strategy of design, function oriented design, object oriented design.
4. **S/W Metrics**
   Introduction, size metrics, data structure metrics, information flow metrics, entropy based measures, metric analysis.
5. **S/W Reliability**
   Importance, S/W reliability & H/W reliability, failures & faults, reliability concepts, reliability models – macro, basic, logarithmic Poisson, calendar time component, micro models; estimating number of residual errors; reliability allocation.
6. **S/W Testing**
   Introduction, Functional testing, structural testing, activities during testing, debugging, testing tools.
7. **S/W Maintenance**
   Introduction, types of maintenance, maintenance process, maintenance models, reverse engineering, re-engineering.

BOOKS

1. R.S. Pressman, Software Engineering – A Practitioner's Approach, 5th Ed, Tata-MGH, 2000.
2. Ian Sommerville, Software Engineering, 4th Ed., Addison Wesley.
3. Pankaj Jalote, An Integrated Approach to Software Engineering 2nd Ed, Narosa Publishing.
4. K.K.Aggarwal, Yogesh Singh: Software Engineering, New Age International Ltd, 2001.

Course Outcome (COs)	Description
CO 1	Understand the correlation between Software Engineering Approach and Process for development of software
CO 2	Apply tools to create prototype using various software development models
CO 3	Analyse various approaches to design, implement and test effective softwares
CO 4	Development of reliable and easily maintainable softwares
CO 5	Demonstrate knowledge of software metrics to design real life software

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	2	1	1			1	1	1	1	2
CO 2	2	2	2	1	3	1		1	2	1	1	1
CO 3	2	3	2	2	2		1	1	2	2		1
CO 4	2	2	3	2	2	2		1	2	2	2	2
CO 5	3	2	3	1	1			1	2	1	2	1

Fundamentals of Database Systems
COT-473
(Open Elective I – For branches except Co, IT)

L	T	P
3	1	-

1. **Basic Concepts**
   What is database system, why database, Data independence, 3 levels of architecture; external level, conceptual level, internal level, mapping DBA, DBMS, organization of databases, components of DBMS, Data Models, Relational Models, Networks data model, Hierarchical Model, semantic data model.
2. **Relational Model**
   Introduction – Relational Model, base tables & views, relations, domains, candidate keys, primary key, alternate keys, foreign key, Integrity rules, relational Operators – relational algebra, relational calculus, tuple calculus, domain calculus, Data Base Design – Introduction, Basic Definitions, Non-loss decomposition and functional dependencies, 1NF, 2NF, 3NF, BCNF, MVD & 4NF, JD & 5NF, Normalization procedure, other normal forms.
3. **Concurrency**
   Transaction concept, transaction state, concurrent executions, serializability lock based protocols, timestamp based protocols, validation based protocols, deadlock handling.
4. **Distributed Data Bases**
   Introduction, fundamental principles, objectives, Problems of distributed processing-query processing, catalog management, updates propagation, recovery control, concurrency control.

BOOKS

1. C.J. Date: An Introduction to Database Systems 7th Ed. Addison Wesley, Indian Edition.
2. A.K. Majumdar and Bhattacharyya: Database Management Systems, THM, 1996.
3. A Silberschatz, H.F. Korth & S. Sudarshan: Data Base System Concepts, MHG, 1977.

Course Outcome (COs)	Description
CO 1	Develop an understanding on the basic concepts, principles and applications of database system.
CO 2	Learn relational database model and SQL
CO 3	Learning database design and normalization techniques
CO 4	To understand and learn transaction management

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2		3	1	2	1	2		1	3	2	1
CO 2	1	2	2	3	3		2			3		3
CO 3	1		2	1	3	2			1	2	3	3
CO 4		2		1	2	3		3	2	3	2	2

Fundamentals of Computer Hardware Technologies
COT-475
(Open Elective I - For branches except Co, IT)

L	T	P
3	1	-

1. **Memory**
   Memory, memory chips & modules, memory types, advanced memory technologies, troubleshooting memory.
2. **Motherboard**
   PC family tree, motherboard controllers and system resources, input-output ports, IRQ, I/O bus system: ISA, MCA, EISA, VESA local bus, PCI, AGP, PCIX; on board I/O devices, ROMBIOS, ROM POST, CMOS setup.
3. **Power Supply**
   Power supply function and operation, power supply quality and specification, power protection and backup, backup power system; UPS; troubleshooting power supply.
4. **Interfaces and I/O Ports**
   Floppy disk interface, IDE interface: ATA standards, master-slave configuration, data transfer mode; SCSI interface: SCSI bus, SCSI standards; which is better SCSI or IDE; serial ports, parallel ports, USB, Video adapters, troubleshooting Video adapters.
5. **Device drives and peripherals**
   Floppy disk drive, hard disk drive, CD ROM drive, DVD ROM drive, record able drives, keyboards, mice, printers and monitor, trouble-shooting drives and peripherals.

BOOKS

1. Craig Zacker & John Rourtre: PC Hardware- The complete reference, TMH.
2. Mark Minosi: The complete PC Upgrade & Maintenance Guide 4/e, BPB publications.
3. Dr. S.K. Chauhan: PC Upgrading, maintenance and troubleshooting guide

Course Outcome (CO's)	Description
CO1	Study all the generation of PC and parts of motherboard, bus system, peripheral devices and storage elements
CO2	Understand the concepts of power supply and backup system
CO3	Knowledge about the different types of device drives, System interfaces and Input Output ports
CO4	Assemble all the parts of motherboard related to hardware and troubleshooting each component of PC

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1		2	1	2			1	1		1	2
CO2			2		2	2	3				2	
CO3					2					1	2	
CO4		2	3		3		3		3		3	3

Artificial Intelligence
COT-477
(Open Elective I – For all branches)

L T P
3 1 -

1. **Introduction**
   Definition of AI, Evolution of Computing, History of AI, Classical, Romantic and Modern period, subject area, Architecture of AI machines, logic family, Classification of logic.
2. **Production System**
   Production rules, The working memory, Recognize-act cycle, conflict resolution strategies, refractoriness, Recency, specificity, alternative approach for conflict resolution by meta rules, Architecture of production system.
3. **Propositional Logic**
   chaining, standard theorems, method of substitution, theorem proving using Wang's algorithm.
4. **Predicate Logic**
   Alphabet of First order logic (FOL), predicate, well formed formula, clause form, algorithm for writing sentence into clause form, Unification of predicates, unification algorithm, resolution Robinson's inference rule, Scene interpretation using predicate logic.
5. **Logic Programming with Prolog**
   Logic program, Horn clause, program for scene interpretation, unification of goals, SLD resolution, SLD tree, flow of satisfaction, controlling back tracking using CUT, common use of CUT, implementation of backtracking using stack, risk of using cuts, fail predicate, application of cut-fail combination, replacing cut-fail by not.
6. **Default and Non monotonic Logic**
   Axiomatic theory, Monotonicity Vs Non-Monotonicity, non-atomic reasoning using McDermott's NML-I, problems with NML-I, reasoning with NML-II, Case study of Truth Maintenance System (TMS), Neural network fundamentals.
7. **Imprecision and Uncertainty**
   Definition, Probabilistic techniques, Certainty factor based reasoning, conditional probability, medical diagnosis problem, Baye's Theorem and its limitations, Bayesian belief network, propagation of belief, Dempster-Shafer theory of uncertainty management, belief interval, Fuzzy relation, inverse Fuzzy relations, Fuzzy post inverse, Fuzzy Inversion.
8. **Intelligent Search Technique**
   Heuristic function, AND-OR graph, OR graph, Heuristic search, A* algorithm and examples.

BOOKS

1. A. Konar: Artificial Intelligence and Soft Computing--Behavioral and Cognitive Modeling of Human Brain, CRC Press, USA.
2. E.Charniak and D. McDermott: Introduction to Artificial Intelligence, Addison Wesley Longman.
3. Elline and Rich: Artificial Intelligence, 2/e 1992.
4. Rich and Knight: Artificial Intelligence, 2/e 1992.

Course Outcome (COs)	Description
CO 1	Acquire fundamental knowledge of various reasoning methods
CO 2	Formulate the problem space in AI for various communication language problems
CO 3	Develop an understanding about heuristics, optimization and complexity
CO 4	Apply knowledge of AI and logical programming using LISP and PROLOG
CO 5	Demonstrate knowledge and understanding of AI techniques to solve live projects

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	3	3	3		3	2	3	2	3	3	3
CO 2	2	3	3	3		1	1			1	2	3
CO 3	1	3	3	2			1		1		2	1
CO 4	1		3	2	1	2	2			2	2	2
CO 5	3	3	3	2	1	2	1			2	2	2

Web Engineering
COT 402

L	T	P
3	1	-

1. Information Architecture

The Role of the Information Architect, Collaboration and Communication, Organizing Information, Organizational Challenges, Organizing Web Sites and Intranets, Designing Navigation Systems, Types of Navigation Systems, Integrated Navigation Elements, Remote Navigation Elements, Designing Elegant Navigation Systems, Grouping Content, Conceptual Design.

2. Dynamic HTML and Web Designing

HTML Basic Concepts, Good Web Design, Process of Web Publishing, Phases of Web Site development, Structure of HTML documents, HTML Elements – Core attributes, Language attributes, Core Events, Block Level Events, Text Level Events, Linking Basics, Linking in HTML, Images and Anchors, Anchor Attributes, Image Maps, Image Preliminaries, Images as Buttons, Introduction to Layout: Backgrounds, Colors and Text, Fonts, Layout with Tables. Advanced Layout: Frames and Layers, HTML and other media types. Audio Support in Browsers, Video Support, Other binary Formats. Style Sheets, Positioning with Style sheets. Basic Interactivity and HTML: FORMS, Form Control, New and emerging Form Elements.

3. CGI using PERL

Introduction to CGI, Alternative Technologies, The Hypertext Transport Protocol, URLs, HTTP, Browser Requests, Server Responses, Proxies, Content Negotiation, The Common Gateway Interface, The CGI Environment, Environment Variables, CGI Output, Forms and CGI, Sending Data to the Server,

4. Java Server Pages

Basics, Integrating Scripts in JSPs, JSP Objects and Components, configuring and troubleshooting, JSP: Request and response objects, Retrieving the contents of a an HTML form, Retrieving a Query String, Working with Beans, Cookies, Creating and Reading Cookies.

5. XML

Relationship between HTML, SGML and XML, Basic XML, Valid Documents, Ways to use XML, XML for Data Files, Embedding XML into HTML documents, Converting XML to HTML for DISPLAY, Displaying XML using CSS and XSL, Rewriting HTML as XML, The future of XML.

Note:- At least one question will be set from each unit.

BOOKS

1. Thomas A Powell, HTML The Complete Reference, Tata McGraw Hill Publications.
2. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e, O'Reilly
3. Pardi, XML in Action, Web Technology, PHI.
4. Yong, XML Step by Step, PHI.

Course Outcome (COs)	Description
CO1	Understand basic principles of web site design, considering the information architecture.
CO2	Incorporate best practices in navigation, usability in website design
CO3	Able to design web pages using HTML tags and CSS.
CO4	Able to demonstrate current Web Technologies(JSP and javascript, CGI with Perl)

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2					1						1
CO 2	2	1	1	1				1	1			1
CO 3	1	2	3	2	1				1		1	
CO 4	2	2	3	2	2				1		1	

Compiler Design
COT-404

L T P
4 1 -

1. Introduction to System Software
Introduction and types of assemblers, linkers, loaders, compilers and translators, the structure of a compiler, different states in the construction of a compiler.

2. Lexical Analysis Design of lexical analyzer.

3 (i) Basic Parsing Techniques
Parsers, shift-reduce parsing, operator- precedence parsing, top-down parsing, predictive parsers.

(ii) Automatic Construction of Efficient Parsers
L.R. Parsers, the canonical collection of L R (O) items, construction of SLR parsing tables, constructing canonical L.R. Parsing tables, Constructing LALR parsing tables, implementation of L R Parsing tables.

4(i) Syntax-Directed Translation
syntax-directed translation schemes, implementation of syntax directed translators, intermediate code, postfix notation, parse trees and syntax trees, three address code, quadruples, and triples, translation of assignment statements, Boolean expressions, control statements.

(ii) Symbol Tables
The contents of a symbol table, data structures for symbol tables, representing scope information.

5 (i) Run Time Storage Administration

Implementation of a simple stack allocation scheme, implementation of block structured languages, storage allocation in FORTRAN storage allocation in block structured language.

(ii) Error Detection And Recovery
Error, Lexical-phase errors, syntactic- phase errors, semantic errors.

6 (i) Introduction to Code Optimization
The principle sources of optimization, loop optimization, the DAG representation of basic blocks, value number and algebraic laws, global data-flow analysis,

(ii) Code Generation
Object programs, problems in code generation, a machine model, a simple code generator, register allocation and assignment, code generation from DAGs, peephole optimization.

BOOKS

1. Aho A.V. and Ullman J.D. Principles of Compiler Design, Addison Wesley.
2. Donovan, J, System Programming, TMH.
3. D.M. Dhamdhere: Compiler Construction-Principles and Practice McMillan India.
4. David Gries: Compiler Construction for Digital Computer.

Course Outcome (COs)	Description
CO1	Learn the fundamentals of language translator and compiler design
CO2	Ability to design algorithms for Parsers (Top-down and Bottom-Up).
CO3	Classify and discuss intermediate code generation and optimization techniques to improve the performance of a program
CO4	Gain knowledge of techniques of symbol table organization, fundamentals of runtime environment and Code generation.

Course Outcomes(COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	1	2	2			1			1			2
CO 2	2	3	3	2	2		1		1	1		2
CO 3	1	2	3	2	1		1			1		2
CO 4	2	2	2	2	2		1					1

Web Engineering (Pr.)
COT-412

L	T	P
-	-	3

1. Chalk out the storyboard and design of Dairy Food Limited. As the name reflects your site provides dairy products and aims at opening an online store. Your storyboard should cover all the features that you plan to have on the site.
2. Create your own page with your favourite hobbies.
3. Create a Menu or a Table of content web page. Each menu item or section of the table of content should load a different web page. For example, if the user clicks on Menu one or section 1 then the link should take him to respective menu html. Or section and so on.
4. Create a web site for your College,
5. Create a frameset that is divided into three sections. The frameset should have three zones.
   - the topmost section of the frameset should take up about just 15% of the browser window. Name this frame title.
   - The middle section should be 70% of the browser window. Name this frame title.
   - The lower most sections should also be about 15% of the browser window. Name this section as menu. Create pages for each section. For the lowermost section, create page that loads the content into the middle section. The topmost section should contain a page describing the web page itself.
6. Create a web page, which displays the map of your Country Link, each city/state on the image using image map, such that the respective HTML page of the city/state is displayed when the user selects an area.
7. Add the tickertape applet to your page by customizing it for the following settings:
   - Increase the count by one.
   - Accordingly update the message count.
   - Change the text color to (237,192,171)
   - Experiment with changing the scrolling speed.
   - Customize the message text as per your page requirement.
8. Incorporate a quest book into the Dairy Food Webpage and use Java Script to build validations into the form.
9. Use Stylesheet to modify the following:
   - Change background to modify the following.
   - Change font, type, face and color.
   - Align Text.
   - Remove underlines from hyperlinks.
10. Use Microsoft's Personal Web Server to set up your website.

Course Outcome (COs)	Description
CO1	Design and validation of forms, web pages using HTML tags and CSS.
CO2	Event-handling on web application and forms.
CO3	Develop a web application using server-side programming languages and component (JSP, javascript)
CO4	Understanding the impact of web designing in the current market place where everyone uses to prefer electronic medium for shopping, commerce, fund transfer and even social life also.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	3		1				1		1	1
CO 2	2	2	3	1					1			
CO 3	2	3	3	1					1			2
CO 4	2	2	3	2	2	2			2		1	2

Distributed Operating Systems
COT-440
(Departmental Elective III)

L	T	P
3	1	-

1. **Architecture of distributed O.S.**
   Introduction, motivation, system architecture type, issues in distributed O.S., Communication primitive.
2. **Distributed mutual Inclusion**
   Introduction, classification preliminaries simple solution, non token based algorithm, Lamport algorithm, Ricart algorithm, Mackawa's algorithm, A generalized non token based algorithm,, token based algorithm, Broad cast algorithm, Heuristic algorithm, tree based algorithm, comparative performance analysis.
3. **Distributed dead lock detection**
   Introduction, dead lock handling strategies, issues in deadlock detection & resolution, Control organization, centralized, distributed & hierarchical detection algorithm.
4. **Distributed file system**
   Introduction, architecture mechanism for building, design issues, log structured file system.
5. **Distributed Scheduling**
   Introduction, motivation, issues in load distribution, component of load algorithm, stabilizing load distribution algorithm, performance comparison, selection of a suitable load sharing algorithm, requirement for load distribution, task migration, issues in task migration.

BOOKS

1. Mukesh Singhal & N.G. Shivaratri: Advanced concepts in operating systems, TMH 2001.
2. A S Tanenbamn: Modern operating systems, PHI.
3. A. Silberschatz, P.Galvin, G.Gagne: Applied operating system concepts, Wiley.

Course Outcome (COs)	Description
CO1	Understanding of architectural details of distributed operating system.
CO2	Design algorithm for mutual exclusion.
CO3	Design distributed deadlock detection and resolution strategies.
CO4	Knowledge of file management system.
CO5	Analysis of load allocation mechanisms.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO12**
CO1	1	2	2	1	1			1	1			1
CO2	2	3	3	2								
CO3	1	3	2	3								
CO4	1	2	1	1	1			1				
CO5	2	3	2	3								

Software Project Measurement
COT-442
(Departmental Elective III)

L	T	P
3	1	-

1. **Software Project Planning**
   Cost estimation models single use multi variable; COCOMO model; COCOMO II model; Putnam model; Delphi model; S/W configuration management; risk management.
2. **Internal and External Attribute Management**
   Size, length, reuse, functionality, complexity, types of structural measures, control flow structures, modularity and information flow attributes, object oriented metrics, data structure, measuring external product attributes- modeling, S/W quality, measuring aspects of quality.
3. **Object-Oriented Design Measurement**
   Cohesion & Coupling Consideration; Coupling – interaction, Component, inheritance; different types of each of these couplings & their measurements; Cohesion- method, class, inheritances; types of each of these Cohesions & their measurements.

BOOKS

1. N.E. Fenton, S.L. Pfleeger: Software Metrics 2/e, Thomson Asia, 2002.
2. Henderson Sellers: Object Oriented Metrics – Measures of Complexity, PH PTR, New Jersy, 1996.
3. K.K.Aggarwal, Yogesh Singh: Software Engineering, New Age International Ltd, 2001.

Course Outcome (COs)	Description
CO 1	Apply model-based formulas to develop software projects
CO 2	Identify and analyze internal attributes to evaluate external quality characteristics
CO 3	Comprehend relationships among different modules to maintain reliable communication
CO 4	demonstrate modern technique to measure the software projects

Course Outcomes (COs)	Programme Outcomes (POs)										
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**
CO 1	3	2	3	2	1	-	1	3	1	2	3
CO 2	3	3	3	2	3	3	2	2	3	3	3
CO 3	2	3	3	3	3	1	2	1	2	3	3
CO 4	3	3	3	2	3	-	2	1	1	3	3

Natural Language Processing
COT 444

L	T	P
3	1	-

1. **Introduction:**
   Language understanding- Phonetic and Phonological aspects of language, Morphological knowledge, Syntax- Semantics and Pragmatics, Articulatory Phonetics, Vocal Organs, Vowel, Syllable, Consonant, Phonetic Transcription, Phonological rules and Transducers.
2. **Grammar and Parsing :**
   Grammar and Sentence structure , Parsing with Context Free Grammar, Top down and bottom up parser, Transition Network Grammars, Morphological Processing , Earley algorithm, Semantic Networks for Knowledge Representation, Types of Semantic Networks.
3. **Probabilistic (Statistical) Models of pronunciation:**
   Dealing with spelling errors, Part of speech Tagging, Bayesian method to spelling and pronunciation, Decision tree models, Computing Likelihood from weighted Automata: Forward Algorithm, Decoding: The Viterbi Algorithm, Probabilistic Context Free Grammars.
4. **N-gram Language Models:**
   Simple N-grams, smoothing, Deleted Interpolation, N-grams for spelling and Pronunciation, Cross entropy for comparing Models, Use of language models in speech recognition, Architecture of Automatic Speech Recognition System

BOOKS

1. Speech and Language Processing By Daniel Jurafsky James H. Martin, Pearson Education.
2. Natural Language Understanding by James Allen, Pearson Education.
3. Spoken Language Processing by X. Huang, A. Acero, H.W. Hon , Prentice Hall New Jersey.

Course Outcome (COs)	Description
CO1	Develop background of translation, information retrieval and statistical discourse processing.
CO2	Introduction to various recent statistical methods in natural language processing.
CO3	Develop familiarity to linguistics and their application to part-of-speech tagging.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	1	1		1	1				1		1
CO2	2	2	2	2	1	1	1					2
CO3	2	2	2	2	2	2	1					2

Bioinformatics
COT-446

(Departmental Elective III)

L	T	P
3	1	-

1. **Introduction to Bioinformatics:**
   Introduction, outline of proteins, primary structure: the 20 amino acids - chemical structure & properties: chirality, different types of side chain: relevance to mutation, size, aliphatic/aromatic, polarity, charge, hydrophobicity; disulphide bonds, molecular models, Polypeptide geometry: the folding chain, nomenclature, molecular graphics
   Structure evolution and mutation genetic information- the triplet code; DNA structure Synthesis of proteins: cell biology background; transcription: RNA polymerase, introns, exons, splicing translation: ribosomes, start/stop codons, Post-translational processing
2. **Computing Evolution:**
   Phylogenetic Analysis Sequence-based taxonomy: Overview and Assumptions, From Multiple Alignment to Phylogeny Neighbor, Joining Maximum Likelyhood vs. Parsimony, The Molecular Clock, Computer Tools for patterns, mapping and phylogenetic analysis.
   Mathematical tools of proteins and nucleic acids, sequence-Function Relationships Sequence Homology and Conserved Regions, Conserved DNA sequences.
3. **Bioinformatics tools:**
   Networks - WWW, CERN EMBnet; EMBL Database, SEQNET, GenBank, NLM, etc., Sequence Databases and Sequence Analysis: genomic, cDNA EMBL database GenBank protein sequence, pattern recognition tools
   Similarity searching, secondary sources, Genome databases, Molecular graphics software and other packages, To find sequences based on keywords & phrases, to grab individual sequences or whole groups of sequences from a database
4. **Genomics:**
   Introduction, genome scale sequencing, comparative and evolutionary genomics, microarrays, proteomics, pharmacogenomics, Development using computer tools for eequencing projects, PCR and restriction mapping, practical and theoretical problems in sequencing, The challenges of whole genome sequencing, Web-based tools for restriction mapping, new technologies and new bioinformatics tools.

BOOKS

1. Teresa K. Attwood, David J. Parry-Smith: Introduction to Bioinformatics, 1999, Longman Higher Education, 0582327881.
2. S. Eddy, A. Krogh, G. Mitchison, Richard Durbin: Biological sequence analysis: probabalistic models of proteins and nucleic acids, 1999, Cambridge University Press. 0521629713
3. Andreas Baxevanis, B.F.Francis Ouellette: Bioinformatics: a practical guide to the analysis of genes and proteins, 1998, John Wiley & Sons, Inc. 0471191965.
4. James D. Tisdall: Beginning Perl for Bioinformatics. 2001. O'Reilly & Associates. 0596000804.

Course Outcome (CO's)	Description
CO 1	Understand the fundamental concepts in bioinformatics
CO 2	Analyze different evolution techniques in bioinformatics
CO 3	Apply different bioinformatics tools to get desired biological information from given database
CO 4	Demonstrate knowledge and understanding to develop real time projects using different bioinformatics methods
CO 5	Ability to design and implement of the new algorithms and be capable of performing simple sequence analyses using existing tools with new algorithm.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	3	1	1	1	1	1	2		1	2	1	2
CO 2	2	2	2	1	1	2	2		1	2	1	2
CO 3	2	3	2	2	3	3	2	1	2	2	1	2
CO 4	3	2	3	2	2	3	2	1	2	1	2	2
CO 5	2	3	3	3	3	3	2	1	2	1	2	2

Software Testing
COT – 450

L	T
3	1

1. **Functional Testing: Boundary Value Testing**: Analysis. robust testing. worst case testing. special & random testing. examples: Equivalence Class Testing equivalence classes. examples: Decision Table Based Testing: decision tables with examples.

2. **Structural Testing**: Path testing: DO-Paths. Metrics. basic testing: Data Flow Testing: DU testing. slice based testing: Mutation testing.

3. **Integration & System Testing**: Levels of testing; integration testing: decomposition based. call graph based & path based integration: System testing: threads based structural & functional testing.

4. **Object Oriented Testing**: Some issues in Object Oriented Testing. Units for object-oriented testing. implications of composition and encapsulation. implication of inheritance. implication of polymorphism. levels of object-oriented testing. data flow testing for object-oriented software.

BOOKS

1. Soris Seizer. .Software Testing Techniques.. Second Edition. Wiley India. 2005.
2. William Perry. -Effective Methods for Software Testing. 3rd Edition. Wiley India. 2006
3. Paul C.Jorgensen. -Software Testing A Craftsman. Approach.. 2nd Ed.. CRC Press. 2002
4. Cern Kaner. Jack Falk. Nguyen Quoc.. -Testing Computer Software.. 2nd Edition. Van Nostrand Reinhold. New York. 1993
5. Louise Tamres_ .Software Testing.. Pearson Education Asia. 2002

Course Outcome (CO's)	Description
CO 1	Analyse different approaches to software testing, and select optimal solutions for different situations and projects with the help of decision tables.
CO 2	Able to define, formulate and analyze a problem using various aspects of Structural, Integration and Object Oriented Testing.
CO 3	Understand the process to be followed in software development life cycle and the role played by software testing in the SLC.
CO 4	Able to Design and Implement a Micro Software Testing project and understand the role of various testing techniques in improving the efficiency and robustness of software.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	2	2	3	1		1				
CO 2	3	3		2	2						1	1
CO 3	1					2	2					
CO 4	3	2	2	3	3	1		2	3	2	3	2

Fundamentals of Operating Systems
COT-472
(Open Elective II – For branches except Co, IT)

L	T
3	1

1. **Introductory Concepts**
   Operating system functions and characteristics, historical evolution of operating systems, Real time systems, Distributed systems, Methodologies for implementation of O/S service system calls, system programs, Interrupt mechanisms.
2. **File Systems**
   Functions of the system, File access and allocation methods, Director Systems: Structured Organization, directory and file protection mechanisms, implementation issues; hierarchy of file and device management.
3. **CPU Scheduling**
   Levels of Scheduling, Comparative study of scheduling algorithms, multiple processor scheduling.
4. **Storage Management**
   Storage allocation methods: Single contiguous allocation, Multiple contiguous allocation, Paging, Segmentation combination of Paging and Segmentation, Virtual memory concepts, Demand Paging, Page replacement Algorithms, Thrashing,
5. **Device Management**
   Hardware Organization, Device scheduling policies
6. **Deadlocks**
   Deadlock characterization, Deadlock prevention and avoidance, Deadlock detection and recovery, practical considerations.
7. **Concurrent Processes**
   Critical section problem, Semaphores, Classical process coordination problems and their solutions, Interprocess Communications.
8. **Protection**
   Mechanisms and Policies, Implementation.

BOOKS

1. Peterson, J.L. & Silberschatz, A.: Operating System Concepts, Addison, Wesley, Reading.
2. Brinch, Hansen: Operating System Principles, Prentice Hall of India.
3. Haberman, A.N.: Introduction to Operating System Design Galgotia Publication, New Delhi.
4. Tanenbaum, A.S.: Operating System.

Course Outcome (COs)	Description
CO1	Understanding of basic principles of operating systems.
CO2	Analysis of file system management.
CO3	Implementation of efficient scheduling algorithms for real-world problems
CO4	Discuss various Storage and Device Management techniques.
CO5	Design of deadlock-free and synchronised concurrent algorithms.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2				1							2
CO 2	1	2	1	1								
CO 3	1	3	3	2								
CO 4	1	1	2	1								
CO 5	1	2	3	3				1	1			

Fundamentals of Computer Networks
COT-474

(Open Elective II – For branches except CO, IT, EC)

L	T	P
3	1	-

1. **Introduction**
   Network Functions, Network Topology, Network Services, Switching Approaches, Transmission media and systems, multiplexing and signaling techniques, Error detection and correction, ISDN and BISDN
2. **Layered Architectures**
   Examples, OSI Reference Model, Overview of TCP/IP architecture, Socket system calls, SNMP, Electronic Mail.
3. **Peer-to-Peer Protocols**
   Protocols, Service Models and End-to-End requirements, ARQ, Sliding Window, RTP, HDLC, PPP protocols, Statistical Multiplexing.
4. **MAC and LAN Protocols**
   Multiple access communication, Random Access-ALOHA, Slotted-ALOHA, CSMA, CSMA-CD, Channelization – FDMA, TDMA, CDMA, Channelization in Cellular networks LAN Standards - 802.3, 802.4, 802.5, 802.6, FDDI, 802.11, LAN Bridges.
5. **Packet Switching Networks**
   Packet network topology, Datagrams and Virtual Circuits – Structure of Switch / Router, Connectionless and Virtual Circuit packet Switching, X.25, Routing Algorithms, Traffic management and QoS – FIFO, Priority Queues, Fair Queuing, Congestion Control techniques.
6. **TCP/IP**
   Architecture, Internet protocols – IP packet, Addressing, Subnet addressing, IP routing, CIDR, ARP, RARP, ICMP, Reassembly, UDP, Transmission Control Protocol – TCP, Reliable stream service, operation, protocol.

BOOKS

1. Leon Garcia and Indra Widjaja: Communication Networks – Fundamental Concepts and Key Architectures, TMH, 2000.
2. A.S. Tanenbaum: Computer Networks, 3/e, PHI, 1997.
3. Forouzan, Coombs and Fegan: Introduction to data Communications and Networks, TMH, 1999.
4. William Stallings: Data and Computer Communications 5/e, PHI.

Course Outcome (COs)	Description
CO 1	Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
CO 2	Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
CO 3	Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
CO 4	Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	2	1			2	2	1				2
CO2	3	2	3	1	1	2	1		1	2	1	2
CO3	2	1	2	2	3	2		2			1	1
CO4	3	3	3	2	2	2	1	1		2		

Object Oriented Software Engineering
(Open Elective II - For all branches)

COT-476

L	T	P
3	1	-

1. Review of Object Oriented Systems

Design Objects, Class hierarchy, inheritance, polymorphism, object relationships and associations, aggregations and object containment, object persistence, meta-classes, Object-oriented systems development life cycle, Software development process Object Oriented systems development: a use-case driven approach.

2. Methodology for Object Oriented Design

Object modeling technique as software engineering methodology, Rumbaugh methodology, Jacobson Methodology, Booch Methodology, Patterns, Frameworks, the unified approach, unified modeling language (UML).

3. Object Oriented Analysis

Analysis Process, Use-Case Driven Object Oriented Analysis, Use-Case Model, Object Classification, Theory, Different Approaches for identifying classes, Classes, Responsibilities and Collaborators, Identifying Object Relationships, Attributes and Methods, Super-sub Class Relationships, A-Part of Relationships-Aggregation, Class Responsibilities, Object Responsibilities.

4. Object Oriented Design

Object oriented design process, corollaries, design axioms, design patterns, object oriented design philosophy, UML Object Constraint Language, Designing Classes: The Process, Class Visibility, Refining Attributes, Designing Methods and Protocols, Packages and Managing classes,

Designing Interface Objects, View layer interface design, Macro and Micro level interface design process.

BOOKS

1. Ali Bahrami, Object Oriented Systems Development,: McGraw Hill, 1999.
2. Rumbaugh et. al., Object Oriented Modeling and Design, PHI, 1997.
3. Grady Booch, Object Oriented Analysis and Design, Addison-Wesley, 1995.
4. Ivar Jacobson, Object Oriented Software Engineering, Addison-Wesley, 1995.
5. Wirfs-Brock, Designing Object-Oriented Software, PHI, 1997.

Course Outcome (COs)	Description
CO1	To understand of object oriented approach and basic aspects related to it
CO2	Able to understand various methodologies and modelling techniques that can be used for object oriented design
CO3	To analyse various approaches to identify objects, classes and their relationship
CO4	Able to apply object oriented designing approach for project development

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1		1	2	2				1		2		2
CO2	1	2	2	3	2	1	1		1		2	3
CO3		2	3	2	2	1	2	2	2	2	3	3
CO4	1		2	1	3	1	3	3	3	3	3	

Expert Systems
COT-478
(Open Elective II – For all branches)

L	T
3	1

1. **Introduction to Expert System**
   Features of expert system, Representation and organization of knowledge, Basic characteristics, Types of problems handled by expert systems, Case study of PROSPECTOR.
2. **Expert System Tools**
   Techniques of knowledge representation in expert systems, knowledge engineering, System-building aids, support facilities, stages in the development of expert systems.
3. **Building an Expert System**
   Expert system development, Selection of tool, Acquiring knowledge, Building process
4. **Problems with Expert Systems**
   Difficulties, common pitfalls in planning, Dealing with domain expert, Difficulties during development.

BOOKS

1. Waterman D.A.: A Guide to Expert Systems, Addison Wesley Longman
2. Hayes-Roth, Lenat and Waterman: Building Expert Systems, Addison Wesley
3. Weiss S.M. and Kulikowski C.A.: A Practical Guide to Designing Expert Systems, Rowman & Allanheld, New Jersey

Course Outcome (COs)	Description
CO 1	Demonstrate knowledge representation techniques to design components of expert system
CO 2	Apply tools and technique for development of expert systems
CO 3	Acquire knowledge of expert systems to formulate new problem
CO 4	Design and implement expert systems for real life problem

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	3	3	1		1		2	1	3	2	1
CO 2	1	1	1	3						2	1	1
CO 3	2	1	3	2	3		1		1	3		
CO 4	3	2	3	2		2	2	3	3	3	2	3