Syllabus

III & IV Semester
Bachelor Degree
in
Computer Science & Engineering

2013-14

P.E.S. College of Engineering

Mandya - 571 401. Karnataka
(An Autonomous Institution Affiliated to VTU Belgaum)
Grant -in- Aid Institution
(Government of Karnataka)
Accredited by NBA, New Delhi
Approved by AICTE, New Delhi.
Ph : 08232- 220043
Fax : 08232 - 222075
Web : www.pescemandya.org

P.E.S. COLLEGE OF ENGINEERING, MANDYA
(An Autonomous Institution)
SCHEME OF TEACHING AND EXAMINATION
III Semester B.E. (CS & E)

Sl No	Course Code	Course Title	Teaching Dept.	hours / week L:T:P:H	Credits	Examination Marks		
						CIE	SEE	Total Marks
1	P13MAT31	Course I - Engineering Mathematics-III	Maths	4:0:0:4	4	50	50	100
2	P13CS32	Digital Logic Design	CS	3:1:0:5	4	50	50	100
3	P13CS33	Data Structures	CS	4:0:0:4	4	50	50	100
4	P13CS34	Discrete Mathematical Structures	CS	4:0:0:4	4	50	50	100
5	P13CS35	Object Oriented Programming with C++	CS	2:1:0:4	3	50	50	100
6	P13CS36	Computer Organization	CS	4:0:0:4	4	50	50	100
7	P13CSL37	Data Structures Lab	CS	0:0:3:3	1.5	50	50	100
8	P13CSL38	Digital Logic Design Lab	CS	0:0:3:3	1.5	50	50	100
9	P13HU39	Aptitude Competence and Professional Augmentation – I (ACPA- I)	HS&M	2:0:0:2	0	(50)	--	--
10	P13xxL310	Industry Interaction - I	CS	0:0:1:1	0	(50)	--	--
11	P13HM311	Constitution of India & Professional Ethics	Human& Science	2:0:0:2	0	(50)	---	---
12	P13HUDIP39	English & Persona Evolution#	HS&M	4:0:0:4	[2]#	[50]#	[50]#	[100]#
13	P13MADIP31	Additional Maths-I	Maths	4:0:0:4	0	(50)	---	---
Total					26[28]	400[450]	400[450]	800[900]

L: Lecture, T: Tutorial, P: Practical, H: Hrs/ Week, CIE: Continuous internal evaluation, SEE semester end Examination, C: Credits
#English & Persona Evolution Lateral entry students shall have to pass these Credit courses before completion of V- Semester.
*Additional Mathematics-I and Constitution of India & professional Ethics Lateral entry students shall have to pass these mandatory learning courses before completion of V- Semester.

Course Code : P13CSL48	**Semester : IV**	**L - T - P : 0:0.5:1**
Course Title : Object Oriented Programming with C++ Laboratory		
No. of Hours per Week: 3, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Student should have

- Problem solving skills and C Language

Course Outcomes:

Students will be able to :

1. Students will be able to design, implement, test, debug, and document programs in C++.
2. Students will be able to solve the real world problems using object oriented concepts.
3. Students will be able to use the rich features provided in programming language to develop solutions to simple problems.
4. Students will be able to identify classes, objects, members of a class and the relationships among them needed for a specific problem.
5. Students will be able to use the object oriented concepts in software development.
6. Students will be able to build good quality software using object-oriented techniques.
7. Students will be able to breakdown simple programming goals into object-oriented components, propose and evaluate different designs for solving problems using knowledge of fundamental programming techniques

Sl. No.	Course Content/Experiments
1	Programs using Class and Objects
2	Programs using Constructors and Destructors
3	Programs on Operator overloading
4	Programs on Inheritance
5	Programs on Polymorphism and Virtual functions
6	Programs on Templates – Function and Class Templates
7	Programs on Exception handling
8	Programs on Stream handling

Course Outcome (CO)	Program Outcome (ABET/NBA-(3a-k))												
	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	l	m
Analysis and Design of Algorithms (P08CS44)	**2**	**2**		**2**	**2**				**2**				**2**
CO1: Develop searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer, transform and conquer technique	2	2	2		2		1		2		2		2
CO2: Implement solutions to the graph based problems using the algorithm techniques such as decrease and conquer, dynamic programming, and greedy technique.	2	2	2		2			1	2		2		2
CO3: Identify and Apply algorithm Techniques to solve realistic problems.	2	2	2						2		2		2
CO4: Design and **implement** algorithms for the given problem.	2	2	2		2		2		2		2		

Course Outcome (CO)	PROGRAMME EDUCATIONAL OBJECTIVES (PEOS)			
	1	**2**	**3**	**4**
Analysis and Design of Algorithms (P08CS44)	**2**	**2**	**2**	**2**
CO1: Develop searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer, transform and conquer technique	2	2	2	
CO2: Implement solutions to the graph based problems using the algorithm techniques such as decrease and conquer, dynamic programming, and greedy technique.		2	2	2
CO3: Identify and Apply algorithm Techniques to solve realistic problems.		2		2
CO4: Design and **implement** algorithms for the given problem.			2	2

P.E.S. COLLEGE OF ENGINEERING, MANDYA
(An Autonomous Institution)
SCHEME OF TEACHING AND EXAMINATION
IV Semester B.E. (CS & E)

Sl No.	Course Code	Course Title	Teaching Dept.	Hours / week L:T:P:H	CrediTS	Examination Marks		
						CIE	SEE	Total Marks
1.	P13MAES41	Course I - Engineering Mathematics-IV (HC)/	Maths	4:0:0:4	4	50	50	100
2.	P13CS42	Graph Theory & Combinatorics	CS	4:0:0:4	4	50	50	100
3.	P13CS43	Theory of Computation	CS	4:0:0:4	4	50	50	100
4.	P13CS44	Analysis and Design of Algorithms	CS	3:1:0:4	4	50	50	100
5.	P13CS45	Unix System Programming	CS	2:1:0:4	3	50	50	100
6.	P13CS46	Microprocessor	CS	4:0:0:4	4	50	50	100
7.	P13CSL47	Analysis and Design of Algorithms Lab	CS	0:0.3:3	1.5	50	50	100
8.	P13CSL48	Object oriented programming with C++ Lab	CS	0:0.3:3	1.5	50	50	100
9	P13HU49	Aptitude Competence and Professional Augmentation – Ii (ACPA- II)	HS&M	2:0:0:2	0	(50)	--	--
10	P13xxL410	Mini Project- I	CS	0:0:1:1	0	(50)	--	--
11	P13MADIP41	Additional Maths-II	Maths	4:0:0:4	0	(50)	--	--
12	P13EV49	Environmental Studies	Env	2:0:0:2	0	(50)	--	--
		Total				400	400	800

: Lecture, T: Tutorial, P: Practical CIE: Continuous internal evaluation, SEE semester end Examination, C: Credits

*Additional Mathematics-II and Environmental Studies Lateral entry students shall have to pass these mandatory learning courses before completion of VI- Semester.

** ACPA – II: All students shall have to pass this mandatory learning courses before completion of VI- Semester

Evaluation Scheme
(For Theory Courses only)

Scheme	Weightage	Marks	Event Break Up				
			Test I	Test II	Quiz I	Quiz II	Assign-ment
CIE	50%	50	35	35	5	5	10
SEE	50%	100	Questions to Set: 10		Questions to Answer: 5		

A. Scheme of SEE Question Paper (100 Marks)

Duration: 3Hrs	Marks: 100	Weightage: 50%

Each of the two questions set shall be so comprehensive as to cover the entire contents of the unit.
There will be direct choice between the two questions within each Unit
Total questions to be set are 10. All carry equal marks of 20
The no of subdivisions in each main question shall be limited to three only
No of questions to be answered by students is 5

Course Content

1. Problems using brute force technique
   - Recursive linear search
   - Selection sort

2. Problems using divide and conquer technique
   - Merge sort
   - Quick sort
   - Recursive binary search

3. Problems using decrease and conquer technique
   - Insertion sort
   - Topological sorting
   - DFS
   - BFS

4. Problems using transform and conquer technique
   - Heap sort

5. Problem on string matching
   - Horspool's algorithm

6. Problems using dynamic programming technique
   - Binomial coefficient
   - Warshall's algorithm
   - Floyd's algorithm
   - Knapsack algorithm

7. Problems using greedy technique
   - Prim's algorithm
   - Kruskal's algorithm
   - Dijkstra's algorithm

8. Problems using backtracking technique.
   - N queens problem

Course Assessment Matrix (CaM)												
Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
Understand the architecture of 8086 microprocessor. (Unit-I)	L 1	2	-	-	-	2	-	-	-	-	-	-
Apply 8086 instruction set for the given problems (Unit-II)	L 3	3	3	1	-	-	2	-	-	-	-	-
Develop different modules & link them. (Unit-III)	L 6	3	3	1	-	2	-	-	-	-	-	-
Apply string instruction set and I/0 Interrupt in 8086 programming (Unit-IV)	L 3	3	3	1	-	2	-	-	-	-	-	-

Course Code : P13CSL47	**Semester : IV**	**L - T - P : 0:0.5:1**
Course Title : Analysis and Design of Algorithms Laboratory		
No. of Hours per Week: 3, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Student should have programming skill in C		

Course Outcomes:

Students will be able to :

1. **Develop** searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer, transform and conquer technique.
2. **Implement** solutions to the graph based problems using the algorithm techniques such as decrease and conquer, dynamic programming, and greedy technique.
3. **Identify and Apply** algorithm Techniques to solve a given contextual problems.
4. **Design** and **implement** algorithms for the realistic problems

Course Code : P13CS32	**Semester : III**	**L - T - P : 3 - 1 - 0**
Course Title : Digital Logic Design		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Subject requires student to know about Basics Electronics		

Course Learning Objectives (CLOs)

This course aims to

1. **Describe** the Boolean Laws and Theorems.
2. **Solve** Boolean expressions using K-map, Q-M and VEM technique.
3. **Design** data processing circuit.
4. **Design** arithmetic circuit .
5. **Analyze** the working of flip-flops .
6. **Use** the flip-flops to design registers and counters.
7. **Design** the asynchronous and synchronous counter for any modulus.
8. **Convert** digital circuit to analog circuit and vice-versa.
9. **Define** VHDL and **write** VHDL code for logic circuits.
10. **Explain** the different logic families and digital integrated circuits.

Course Content

Unit-1

Digital Logic and Combinational Logic Circuits:
Overview of Basic Gates and Universal Logic Gates, AND-OR –Invert Gates, Positive and Negative Logic, Boolean Laws and Theorems, Sum-of-products Method, Truth table to Karnaugh Map, Pairs, Quads, and Octets, Karnaugh Simplification, Don’t Care Conditions, Product-of-Sum Method, Product-of-sum Simplification, Simplification by Quine-McClusky Method, Simplification by VEM Technique 10 Hours

Unit-2

Data Processing Circuits and Arithmetic Circuits:
Multiplexers, Demultiplexers, 1-of-16 Decoders, BCD-to-Decimal Decoders, Seven-segment Decoders, Encoders, Ex-OR gates, Parity Generators and Checkers, Magnitude Comparators, Read-only memory, Programmable Array Logic, Programmable Logic Array, Design of code converters, Half Adder, Full Adder, Half Subtractor, Full Subtractor, Fast Adder, Adder- Subtractor 10 Hours

Unit-3

Flip-Flops and Simple Flip-Flop Applications: The Basic Bistable Elements, Latches, Timing Considerations, Master-Slave Flip-Flops Pulse-triggered Flip-Flops, Edge – Triggerred Flip-Flops, Characteristics Equations, conversion of flip-flops, Registers: types of registers, serial in serial out , serial in parallel out, parallel in serial out, parallel in parallel out, Application of shift registers: Ring counter, Johnson counter, sequence detector and sequence generator. 11 Hours

Unit-4

Asynchronous and synchronous counter: Asynchronous counter- up, down, up and down counter, design of synchronous up counter and down counter, decade counter, counter design as a synthesis problem.

D/A Conversion and A/D Conversion: Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-slope A/D Conversion, A/D Accuracy and Resolution 11 Hours

Unit-5

VHDL Programming: Introduction to VHDL, Describing data flow, Behavioral, Structural and Mixed design style, Simulating design for Arithmetic and Combinational circuits.

Digital Integrated Circuits: Switching Circuits,TTL Parameters, TTL Overview,Three-state TTL Devices,74C00 CMOS, CMOS Characteristics.

10 Hours

TEXT BOOKS:

1. Digital Principles and Applications: Donald P Leach, Albert Paul Malvino & Goutham Saha, TMH, 7th Edition, 2006.
2. A Verilog HDL Primer, 2nd Edition, J . Bhaskar, BS Publications

REFERENCE BOOKS:

1. Digital Principles & Design by Donald D Givone, 4th Reprint, Tata McGraw Hill 2009.
2. Fundamentals of Digital Logic with Verilog Design, Stephen Brown, ZVonkoVranesic ,TMH, 2006

Course Outcomes

After learning all the units of the course, the student is able to

1. **Understand** Basic gates and Universal gates .
2. **Design** the logic circuit
3. **Apply** the knowledge of number system.
4. **Design** the counters using flip-flops.
5. **Design** the shift registers.
6. **Understand** and **Write** the VHDL code for all logic circuits.

Topic Learning Objectives

After learning all the topics of unit – I, the student is able to

Topic Learning Objectives (Unitwise)

Sl. No.	Topic	Learning Objective
	Unit I	

Unit-4

7. Interrupt I/O
8. Interrupt I/O (Contd…)
9. Block transfers and DMA
10. Block transfers and DMA (Contd…)

Unit-5

1. Basic 8086/8088 configurations
2. Basic 8086/8088 configurations (Contd…)
3. Basic 8086/8088 configurations (Contd…)
4. Basic 8086/8088 configurations (Contd…)
5. System Bus Timing
6. System Bus Timing (Contd…)
7. Interrupt Priority Management
8. Interrupt Priority Management (Contd…)
9. Bus Standards
10. Bus Standards (Contd…)

A. Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
Understand the architecture of 8086 microprocessor. (Unit-I)	L 1	M	-	-	-	M	-	-	-	-	-	-
Apply 8086 instruction set for the given problems (Unit-II)	L 3	H	H	L	-	-	M	-	-	-	-	-
Develop different modules & link them. (Unit-III)	L 6	H	H	L	-	M	-	-	-	-	-	-
Apply string instruction set and I/0 Interrupt in 8086 programming (Unit-IV)	L 3	H	H	L	-	M	-	-	-	-	-	-
Understand min & max mode of 8086. (Unit-V)	L 1	M	-	-	-	-	-	-	-	-	-	-

L- Low, M- Moderate, H-High

Lesson Plan

Unit-1

1. Introduction
2. CPU architecture
3. CPU architecture (Contd…)
4. Internal Operation
5. Machine Language Instructions
6. Machine Language Instructions (Contd…)
7. Instruction formats
8. Instruction formats (Contd…)
9. Instruction Execution Timing
10. Instruction Execution Timing (Contd…)

Unit-2

1. Assembler instruction format
2. Assembler instruction format (Contd…)
3. Data transfer Instructions (Contd…)
4. arithmetic instructions
5. arithmetic instructions (Contd…)
6. branch & loop instructions
7. branch & loop instructions (Contd…)
8. logical Instructions
9. directives and operators
10. directives and operators (Contd…)

Unit-3

1. Modular Programming - Linking and Relocation
2. Modular Programming - Linking and Relocation (Contd…)
3. Access to External Identifier
4. Access to External Identifier (Contd…)
5. Stacks
6. Stacks (Contd…)
7. Procedures
8. Procedures (Contd…)
9. Interrupts and Interrupt Routines
10. Interrupts and Interrupt Routines (Contd…)
11. MSAM Macros

Unit-4

1. String Instructions
2. String Instructions (Contd…)
3. Fundamental I/O considerations
4. Fundamental I/O considerations (Contd…)
5. Programmed I/O
6. Programmed I/O (Contd…)

1	Digital Logic and Combinational Logic Circuits	1. Understand basic gates and universal gates. 2. Write Boolean equations for logic circuits and draw circuits for Boolean equations. 3. Use DeMorgan’s first and second theorems to create equivalent circuits. 4. Explain the Karnaugh map method to minimize the expression . 5. Produce minimal expressions from incomplete Boolean expressions. 6. Determine the minimal prime implicates using QM method. 7. Design the logic circuit using basic gates. 8. Construct the logic circuit using universal gates. 9. Produce the minimal expression using VEM technique.
		Unit II
2	Data Processing and Arithmetic Circuits	1. Describe the half-adder, full-adder and adder-subtractor. 2. Design a fast adder circuit that user parallelism to speed up the responses. 3. Describe to convert code from BCD to excess-3 code and BCD to seven segment code. 4. Describe to convert code from BCD to gray code. 5. Explain how a Magnitude Comparator works. 6. Design n bit Magnitude Comparator. 7. Explain the purpose of parity checking. 8. Determine the output of a multiplexer or de-multiplexer based on input conditions. 9. Describe a ROM, PROM, EPROM, PAL and PLA.

Unit III		
3	Flip-Flops and Simple Flip-Flop Applications	1. State the purpose of a clock in a digital system and demonstrate an understanding of basic terms and concepts related to clock waveforms. 2. Describe characteristic equations of Flip-Flops and analysis techniques of sequential circuit. 3. Describe excitation table of Flip-Flops and explain conversion of Flip-Flops as synthesis example. 4. Describe the operation of the basic RS flip-flop and explain the purpose of the additional input on the gated RS flip-flop. 5. Show the truth table for the edge triggered ,RS flip-flop, edge-triggered D flip-flop, and edge triggered JK flip-flop. 6. Discuss some of the timing problems related to flip-flop. 7. Draw a diagram of a JK master-slave flip-flop and describe its operation. 8. Demonstrate the different types of registers. 9. Analyze the applications of shift registers. 10. Explain Ring counter, Johnson counter and sequence detector.

Unit IV		
4	Asynchronous and synchronous counter D/A Conversion and A/D Conversion	1. Define counter. 2. Differentiate between asynchronous and synchronous counter. 3. Describe the basic construction and operation of an asynchronous counter. 4. Construct the up, down and up-down asynchronous counter. 5. Describe the synchronous counter and its advantages. 6. Construct the up, down and up-down synchronous counter. 7. See how the modulus of a counter can be reduced by skipping one or more of its natural counts..

Unit V		
1.	Basic 8086/8088 configurations	Explains basic 8086/8088 configurations
2.	System Bus Timing	Describe System Bus Timing
3.	Interrupt Priority Management	Discuss Interrupt Priority Management
4.	Bus Standards	Understand Bus Standards

Review Questions

1. Describe the memory map of a PC system, with a neat diagram.
2. Explain the flags of 8086 processor using suitable example.
3. What are the advantages of memory paging? Illustrate the concept of paging with neat diagram
4. Discuss the following addressing modes with examples: 1. Direct
 2. Register indirect 3. Base plus index 4. Immediate 5. Scaled indexed
5. Describe the following instruction with suitable examples:
i) PUSH ii) MUL iii) IN iv) AAA
6. Write an ALP using 8086 instructions to generate and add the first l0 even numbers and save the numbers and result in memory location Num and Sum.
7. Bring out the importance of XLAT instruction using a suitable program.
8. Write an ALP using 8086 instructions to count the numbers of zeros in a given 8 bit number and store the result in memory location 'Res'.
9. Explain the following assembler directives: i) Assume; ii) Proc iii) Ends iv) DB.
10. Briefly explain any four bit test instructions.
11. Explain public and extrn directives of assembler and write ALP to read data through keyboard using external procedure and save the keycode in public data segment
12. Write a C program that uses '-asm' function to display strings on output device.
13. Explain in brief the functions of 8086 pins.
14. Describe demultiplexing of multiplexed AD bus with neat diagram.
15. With neat timing diagram, explain memory read cycle.
16. Interface 512 KB RAM to 8088 MP using 64 KB RAM using 3:8 decoder with starting address of memory as 80000H. Clearly mention decoding logic and memory map
17. Explain memory bank selection in 8086 and mention the number of memory bank in 80x86 MPs.
18. Differentiate between memory mapped I/O and I/O mapped I/O (isolated I/O)

Topic Learning Objectives (Unit wise)		
Sl. No.	**Topic**	**Learning Objective**
	Unit II	
1.	Assembler instruction format	Understand Assembler instruction format
2.	Data transfer Instructions	Analyze Data transfer Instructions
3.	arithmetic instructions	Discuss arithmetic instructions
4.	branch & loop instructions	Determine branch & loop instructions
5.	logical Instructions	Explain logical Instructions
6.	directives and operators	Discuss about directives and operators
	Unit III	
1.	Modular Programming - Linking and Relocation	Understand the concept of Linking & Relocation of Modular Programming
2.	Access to External Identifier	Describe Access to External Identifier
3.	Stacks	Understand about Stacks
4.	Procedures	Understand about Procedures
5.	Interrupts and Interrupt Routines	Explains Interrupts and Interrupt Routines
6.	MSAM Macros	Explains MSAM Macros
	Unit IV	
1.	String Instructions	Describe String Instructions and Solve it
2.	Fundamental I/O considerations	Explains Fundamental I/O considerations
3.	Programmed I/O	Understand Programmed I/O
4.	Interrupt I/O	Explains Interrupt I/O
5.	Block transfers and DMA	Understand Block transfers and DMA

	Unit IV	
4		8. Define accuracy and resolution. 9. Design ladder networks for analog to digital conversion.
	Unit V	
5	VHDL Programming and Digital Integrated Circuits	1. Define VHDL. 2. List types of VHDL Programming tecniques. 3. Describe dataflow model. 4. Explain behavioral model. 5. Explain Structural model. 6. Describe Mixed design style. 7. Write the VHDL code for arithmetic and combinational circuit. 8. Explain how diodes and transistors can be used as electronic switches. 9. Demonstrate an understanding of TTL devices, their parameters and how to drive them. 10. Be familiar with CMOS devices and characteristics.

Review Questions

UNIT - 1

1 What is a universal gate? Consider a gate which takes two inputs A and B and produces an output A’.B. Would you consider it a universal gate? Discuss

2 Implement F = (CD+E) (A+B’) using NAND gates.

3 Verify the following Boolean algebraic manipulation. Justify each step with a reference to a postulate or theorem :

(X + Y’ + XY) (X + Y’) X’Y = 0
(AB+C+D) (C’ +D) (C’ + D+E) = ABC’ + D

4 Convert the given expression in standard SOP form F(A,B,C) = A+AB+CB

5 Convert the given expression in standard POS form F(P,Q,R)=(P+Q’) (P+R)

6 Represent each of the following Boolean functions on a Karnaugh map.

a. F(w,x,y,z) = w’ x’ y z’ + w’ x’ y z + w’ x y’ z’ + w’ x y ‘ z +w x’ y z’ + w x y’ z

F(x,y,z) = (x+z) (y+z) (y’ + z’)

7 Apply Karnaugh map technique to the following Boolean functions and simplify

F(A,B,C,D) = A’ B’ C + AD +BD’+CD’ +AC’ +A’B’
F(A,B,C,D) = πM(1,2,4,5,7,8,10,11,13,14) + d(0,3,6,12)

8 Use Quine- McClusky tabulation method and simplify the following functions.

F(a,b,c,d) = ∑ m (0,1,2,3,8,9)
F(p,q,r,s) = ∑ m(0,1,4,5,9,10,12,14,15) +∑ d(2,8,13)

9 Explain the procedure for loading a K-map using map entered variable technique with an example.

10 Simplify the following using MEV technique. Reduce 4-variable to 3-variable.

F(w,x,y,z) = ∑m(2,4,5,10,11,13) + ∑ d(0,1,6,15)
F(p,q,r,s) = πM(1,2,3,7,8,9,14,15)

UNIT -2

11 Realize a full adder using minimum number of two input NAND gates. Write the relevant expressions, truth table and logic diagram.

12 Draw and explain the block diagram of n-bit parallel adder.

13 Realize a full subtractor using basic gates only.

14 What is a high speed adder? Design a 4 bit carry look ahead adder circuit.

15 Design
a. BCD to Excess – 3 code converter
b. Binary to gray code converter

16 Design and explain two bit magnitude comparator.

17 Implement the following function using 8:1 multiplexer
F(w,x,y,z) = ∑ m (0,1,5,6,8,10,12,15)

18 What is decoder? Using gates, show how do you design a 3-to-8 line decoder.
i) Explain a decimal to binary encoder using four OR gates. What is a priority encoder?

19 Design the logic circuit for odd parity checker.
Implement the following Boolean expressions using PROM
f1(a,b,c) = ∑m(0,2,4,7) f2(a,b,c) = ∑m(1,3,5,7)
Implement the following Boolean functions using PLA
F1(a,b,c) = ∑m(0,1,3,5) F2(a,b,c)= ∑m(0,3,5,7)

UNIT - 3

20 Define clock cycle time? Explain with neat waveform.

21 State and explain the characteristics of ideal clock waveform.

22 Explain the working of S-R flip-flop by using NOR gates only.

23 Define race around condition? Explain how it is eliminated.

24 Explain the operation of the master- slave JK flip-flop along with a circuit diagram.

25 Derive the characteristic equation of various flip-flops.

26 Convert i) SR flip-flop to JK flip-flop ii) JK flip-flop to T flip-flop

27 Give the logic diagram of 4bit bidirectional shift register with parallel load capability and briefly explain its operations

Unit-5

System Bus Structure : Basic 8086/8088 configurations – Minimum mode, Maximum mode, System Bus Timing, Interrupt Priority Management – Interrupt System based on Single 8259A, Interrupt System Based on Multiple 8259As, Bus Standards 10 Hours

Text Book:

1. **Microprocessor Systems: The 8086/8088 Family,** Glenn A.Gibson, Prentice-Hall of India, 2nd edition, 1986.

Reference Books :

1. **The Intel Microprocessors**, Barry.B.Brey, PHI Publication, 8th edition, 2009
2. **Microprocessor and Interfacing,** Douglas V.Hall, TMH, 2nd edition 2006.
3. **The Intel Microprocessor Family: Hardware and Software Principles and Applications,** James L. Antonakos, Thomson, 2007.

Course Outcomes:

1. **Understand** the architecture of 8086 microprocessor. (Unit-I)
2. **Apply** 8086 instruction set for the given problems (Unit-II)
3. **Develop** different modules & link them. (Unit-III)
4. **Apply** string instruction set and I/0 Interrupt in 8086 programming (Unit -IV)
5. Understand min & max mode of 8086. (Unit-V)

Topic Learning Objectives (Unit wise)

Sl. No.	**Topic**	**Learning Objective**
	Unit I	
1	Introduction	Understand the introduction about general Microprocessor.
2.	CPU architecture	Explain 8086 CPU architecture.
3.	Internal Operation	Analyze the internal operation of 8086 Microprocessor.
4.	Machine Language Instructions	Determine Machine Language Instructions
5.	Instruction formats	Apply instruction formats.
6.	Instruction Execution Timing	Discuss Instruction Execution Timing

Course Code : P13CS46	**Semester : IV**	**L - T - P : 4-0- 0**
Course Title : Microprocessor		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Student should have knowledge of Computer Organization		

Course Learning Objectives (CLOs)

This course aims to

1. Study in-depth the hardware and software included in micro computer systems.
2. Analyze the concept considered are general in nature, the discussion is based on the particular microprocessor, the Intel 8086/8088 and its associated supporting devices and software.
3. Learn the programming concept of 8086 programming using (Microsoft assembler) MSAM Assembler.
4. Understand 32/64 bit architectures supporting pipelined and superscalar architectures.

Course Content

Unit-1

8086 Architecture: CPU architecture, Internal Operation, Machine Language Instructions addressing modes, Instruction formats, Instruction Execution Timing 10 Hours

Unit-2

Assembler language programming : Assembler instruction format, Data transfer Instructions, arithmetic instructions, branch instructions- conditional branch instruction, unconditional branch instructions, loop instructions, NOP and HLT instructions, logical Instructions, Shift and Rotate Instructions, directives and operators- data definition and storage allocation, structure, records, assigning name of expression, segment definition, program termination, alignment directives, value returning Attribute Operators 11 Hours

Unit-3

Modular Programming : Linking and Relocation – Segment Combination, Access to External Identifiers, Stacks, Procedures – Calls, Returns and Procedure Definitions, Saving and Restoring Register, Interrupts and Interrupt Routines, MSAM Macros 11 Hours

Unit-4

Byte and String Manipulation: String Instructions, REP prefix, table translation.

I/O programming: Fundamental I/O considerations, Programmed I/O, Inter I/O, Block transfers and DMA 10 Hours

UNIT-4

28 Sequence

0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0

Design a counter that has a repeated sequence of 6 states listed above. Give its state diagram

29 Explain with the help of neat diagram 4 – bit asynchronous decade counter.

30 Design an excess -3 decimal counter using JK flip-flops.

31 Differentiate between synchronous and asynchronous counter.

32 Define the following terms for D/A converters:
i. Resolution ii. Accuracy iii. Monotonicity iv. Conversion time

33 Obtain an expression for the output voltage of R/2R DAC.
i. An 8-bit DAC has an output voltage range of 0 – 2.55 V. Define its resolution in two ways.
ii. Find out step size and analog output for 4-bit R-2R ladder DAC when input is 1000 and 1111. Assume v_{ref} = +5V.

UNIT-5

34 Define VHDL? Write the VHDL code 8:1 multiplexer. Define VHDL?

35.Write the VHDL code 8:1 multiplexer.

36 Write the VHDL code for full adder and full subtractor using data flow model.

37 Apply Behavioral model and write the VHDL code for D flip-flop.

38 Write the VHDL code for JK flip-flop using data flow model.

39 Write the VHDL code for ring counter and Johnson counter.

40 Explain Semiconductor diodes.

41 Define MOSFET.

42 Analyze TTL parameters.

43 Explain CMOS characteristics.

44 Design CMOS NAND and NOR gate

Lesson Plan (Unit wise)

UNIT-1

1. The Basic gates- NOT, OR, AND and Universal gates.
2. AND-OR –Invert Gates, Positive and Negative Logic, Boolean Laws and Theorems.
3. Truth table to Karnaugh Map, Pairs, Quads, and Octets.
4. Karnaugh Simplification.
5. Don’t Care Conditions, Product-of-Sum Method.

6. Product-of-sum Simplification.
7. Simplification by Quine-McClusky Method.
8. Simplification by VEM Technique.
9. Problems on simplification of Boolean expressions using Boolean laws and theorems.
10. Problems on Simplification of Boolean functions using K-map, Q-M an VEM technique

UNIT-2

1. Multiplexers, Demultiplexers.
2. 1-of-16 Decoders, BCD-to-Decimal Decoders
3. Seven-segment Decoders, Encoders
4. Ex-OR gates, Parity Generators and Checkers
5. Magnitude Comparators
6. Read-only memory, Programmable Array Logic
7. Programmable Logic Array, Design of code converters
8. Design of code converters
9. Half Adder, Full Adder ,Half Subtractor, Full Subtractor
10. Fast Adder, Adder- Subtractor

UNIT-3

1. The Basic Bistable Elements, Latches
2. Timing Considerations
3. Master-Slave Flip-Flops
4. Pulse-triggered Flip-Flops
5. Edge – Triggerred Flip-Flops, Characteristics Equations
6. Conversion of flip-flops
7. Registers: types of registers, serial in serial out
8. Serial in parallel out, parallel in serial out, parallel in parallel out
9. Application of shift registers: Ring counter
10. Johnson counter, sequence detector and sequence generator
11. Exercises on Flip-flops and registers.

UNIT-4

1. Asynchronous counter- up, down counter
2. Asynchronous up and down counter
3. Design of synchronous up counter and down counter
4. Decade counter
5. Counter design as a synthesis problem
6. Variable, Resistor Networks
7. Binary Ladders, D/A Converters
8. D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion
9. A/D Converter-Counter Method, Continuous A/D Conversion
10. A/D Techniques
11. Dual-slope A/D Conversion, A/D Accuracy and Resolution.

Understand the Shell Programming concepts. (Unit-III)	L 3	**M**	-	-	-	-	-	-	-	-	-	-			
Apply the knowledge of Shell Programming concepts to write Shell Programming examples. (Unit-III)	L 3	**H**	**H**	**L**	-	**M**	-	-	-	-	-	-			
Identify the basic principles of UNIX system program in high level O.S. structure. (Unit -IV)	L 5	**M**	-	-	-	-	-	-	-	-	-	-			
Associate the UNIX file APIs with UNIX system program. (Unit-IV)	L 3	**M**	-	-	-	-	-	-	-	-	-	-			
Understand the UNIX process architecture & control (Unit-V)	L 5	**H**	**H**	**L**	-	-	-	**M**	-	-	**L**	-			
L- Low, M- Moderate, H-High															
Fig 3. Illustration of CAM of Unix System Programming															

6. exec Functions
7. Changing User IDs and Group IDs, Interpreter Files, system Function
8. Process Accounting, User Identification, Process Times, I/O Redirection
9. Process Relationships: Introduction, Terminal Logins, Network Logins
10. Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs, Orphaned Process

Course Articulation Matrix (CAM)															
Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))													
		a	b	c	d	e	f	g	h	i	j	k	l	m	n
Describe UNIX operating system operations and operating system structures. (Unit-I)	L 4	**M**	-	-	-	**M**	-	-	-	-	-	-			
Understand UNIX O.S. file system & File handling commands. (Unit-I)	L 5	**H**	**H**	**L**	-	-	**M**	-	-	-	-	-			
Understand UNIX O.S. file attributes (Unit-II)	L 5	**H**	**H**	**L**	-	**M**	-	-	-	-	-	-			
Understand UNIX O.S. process management & Recognize process handling commands of UNIX (Unit-II)	L 3	**M**	-	-	-	-	-	-	-	-	-	-			

UNIT-5

1, Introduction to VHDL. Describing data flow style with example
2. Behavioral style with example
3. Structural and mixed design style with example
4. Simulating design for arithmetic and combinational circuits
5. Switching circuits
6. TTL Parameters, TTL Overview
7. Three state TTL Devices
8. 74C00 CMOS
9. CMOS Charactersistics
10. Exercises.

Course Articulation Matrix (CAM)												
Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
Understand Basic gates and Universal gates .	**L1**	M	-	-	-	M	-	-	-	-	-	-
Design the logic circuit	**L3**	H	**H**	L	-	-	M	-	-	-	-	-
Apply the knowledge of number system.	**L4**	H	H	L	-	-	M	-	-	L	-	-
Design the counters using flip-flops.	**L4**	H	H	L	-	M	-	-	-	-	-	-
Design the shift registers.	**L4**	H	H	L	-	-	M	-	-	L	-	-
Understand and **Write** the VHDL code for all logic circuits.	**L1 & L4**	H	M	H	-	-	-	M	-	-	L	-
L- Low, M- Moderate, H-High												
Fig 3. Illustration of CAM of Digital Circuits Design												

Course Assessment Matrix (CaM)

Course Outcome (CO)		**Program Outcome (ABET/NBA-(3a-k))**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Understand Basic gates and Universal gates .	**L1**	2	-	-	-	2	-	-	-	-	-	-
Design the logic circuit	**L3**	3	**3**	1	-	-	2	-	-	-	-	-
Apply the knowledge of number system.	**L4**	3	3	1	-	-	2	-	-	1	-	-
Design the counters using flip-flops.	**L4**	3	3	1	-	2	-	-	-	-	-	-
Design the shift registers.	**L4**	3	3	1	-	-	2	-	-	1	-	-
Understand and **Write** the VHDL code for all logic circuits.	**L1 & L4**	3	2	3	-	-	-	2	-	-	1	-

1 – Low, 2 – Moderate and 3 – High

Fig 5a. Illustration of CAM of Digital Circuits Design

8. Shell Programming Examples
9. Shell Programming Examples
10. Shell Programming Examples

Unit-4

1. UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards
2. UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards
3. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.
4. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.
5. General File APIs, File and Record Locking, Directory File APIs, Device File APIs
6. General File APIs, File and Record Locking, Directory File APIs, Device File APIs
7. FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files
8. FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files
9. dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program
10. dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program

Unit-5

1. The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program
2. The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program
3. Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes
4. Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes
5. Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions

File Ownership

3. File Systems and Inodes, Hard Links, Symbolic Links and ln, The Directory, umask: Default File and Directory Permissions, Modification and Access Times. find: Locating Files, Converting One File to Other, dos2unix and unix2dos: Converting between DOS and UNIX, Compressing Files, gzip, gunzip, zip and unzip commands, tar command.
4. Process Basics, ps: Process Status, System Processes, Mechanism of Process Creation, Internal and External Commands, Running Jobs in Background
5. nice: Job Execution With Low Priority, Killing Processes with Signals, Job Control, at and batch: Execute Later, cron: Running Jobs Periodically, time: Timing Processes
6. The Sample Database, pr: Paginating Files, head: Displaying the Beginning of a File
7. tail: Displaying the End of a File, cut: Slitting a File Vertically, paste: Pasting Files
8. sort: Ordering a File, uniq: Locate Repeated and Non repeated Lines, tr: Translating Characters, An Example: Displaying a Word-count List
9. Examples of Simple Filters
10. Revision

Unit-3

1. The Shell's Interpretive Cycle, Pattern Matching The Wild-cards, Escaping and Quoting, Redirection: The Three Standard Files, /dev/null and /dev/tty:Two Special Files, Pipes
2. tee: Creating a Tee, Command Substitution, Shell Variables, Environment Variables, Aliases (bash and ksh), Command History (bash and ksh)
3. Shell Scripts, read and read-only commands, Using Command Line Arguments, exit and Exit Status of Command, The Logical Operators && and || -Conditional Execution, The if Conditional, Using test and [ ] to Evaluate Expressions
4. The case Conditional, expr: Computation and String Handling, $0: Calling a Script by Different names, while: Looping, for: Looping with a List, set and shift: Manipulating the Positional Parameters
5. The here Document (<<), trap: Interrupting a Program, Debugging Shell Scripts with set -x, export: Exporting Shell Variables, eval: Evaluating Twice, The exec Statement
6. Development of simple shell scripts to demonstrate the integer and real arithmetic operations, handling of positional parameters, the use of branching and looping constructs in the shell, handling of signals using the trap etc.
7. Shell Programming Examples

Course Code : P13CS33	**Semester : III**	**L - T - P : 4 - 0 - 0**
Course Title : Data structure with C		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Subject requires student to know about
1. Basic C programming skills
2. Basics of computers

Course Learning Objectives (CLOs)

This course aims to

1. **Analyze** the need for data structuring techniques,
2. **Design** and Implement standard data structures like stack using recurssion .
3. **Learn** the different types of linked list
4. **Design** and implement operations on SLL, DLL, Circular SLL and Circular DLL using header nodes.
5. **Learn** the Basic operations on - Linear queue, Circular queue, Priority Queue and Double ended Queue .
6. **Design a**nd Implement different types of queues Using SLL.
7. **Identify** the différent tree traversal techniques
8. **Design and** implement different tree traversal techniques using iteration and recursion.
9. **Learn** the different sorting and searching techniques.
10. **Analyze** the performance of the different sorting and searching techniques.. .

Course Content

Unit-1

Introduction to data structures-Definition, Abstract Data Types-ADT for rational numbers, ADT for varying length Character String**,** Classification of Data Structures.

Stacks

Representing stack in C- Implementation of Push, Pop and display operations using arrays and pointers.

Example of Stacks: Infix, Postfix, Prefix, Infix to postfix, prefix to postfix, evaluation of postfix.

Recursion

Definition ,Writing Recursive programs-Factorial Numbers, Fibonacci Numbers and Tower of Hanoi Problem 10 Hours

Unit-2

Linked Lists

Static Memory Allocation and Dynamic Memory Allocation, Basic operations on SLL, DLL, Circular SLL and Circular DLL: insertion, deletion and display. Implementation of SLL with Header nodes 10 Hours

Unit-3

Applications of Linked Lists: Merging, Reversing, Searching, Addition of two polynomials using SLL.

Queues

Definition, Representation, operations, implementation using arrays and linked lists. Different types of queues, Basic operations on - Linear queue, Circular queue, Priority Queue and Double ended Queue (Using SLL), Applications of Queues 10 Hours

Unit-4

Trees

Introduction-Definition, Tree Representation, Properties of Trees, Operations on Binary tree, Binary Search Tree [BST] - Definition, searching BST, Insertion to BST, Deletion from BST, Display BST

Tree and their Applications- Tree Traversal, General Expression as a tree, Evaluating an Expression Tree; Threaded Binary Trees-Threads, Inorder Traversal of a Threaded Binary Tree, Inserting a Node into a Threaded Binary Tree 12 Hours

Unit-5

Sorting Techniques

Insertion sort, Quick sort, Binary tree sort, Heap sort, Merge sort.

Searching Techniques

sentinel search, probability search, ordered list search (Text Book-2) 10 Hours

Text Book :

1. Data Structures using C and C++ by Yedidyah Langsam and Moshe J. Augenstein and Aaron M.Tenanbaum, PHI, 2nd Edition.
2. Data Structures – A pseudo code Approach with C – Richard F Gilberg and Behrouz A forouzan, 2nd Edition.

Reference Book :

1. Fundamentals of Data Structures in C - Horowitz, Sahani, Anderson-Freed, Second Edition, University Press, 2nd Edition. Understand primitive and derived data structure.

Course Outcomes

After learning all the units of the course, the student is able to

1. Understand Abstract data types, Stacks and recursion.

12. Difference between wait3 and wait4
13. Explain exec function
14. Explain Race Condition
15. Explain Process Accounting
16. Explain Job Control
17. What are abnormal terminations
18. What is process group and session
19. What are the different attributes and explain each
20. Explain process termination
21. Explain process creation
22. Explain file sharing
23. Explain hard link and symbolic link
24. What are the different types of files and how they are created

Lesson Plan

Unit-1

1. Brief history, Salient features of a UNIX System, The UNIX Architecture. Introduction to Linux Operating System, Internal and External Commands
2. Introduction to system administration, man: Browsing and Manual Pages On-line, cal: The Calendar, date: Displaying and System Date, echo: Displaying a Message, printf: An Alternative to echo
3. bc: The Calculator, script: Recording Your Session, passwd: Changing Your Password, who, uname: Knowing Your Machine's Characteristics
4. tty: Knowing Your Terminal, stty: Displaying and Setting Terminal Characteristics
5. The File, The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd: Checking Your Current Directory,
6. cd: Changing the Current Directory, mkdir: Making Directories, rmdir: Removing Directories, Absolute Pathnames, Relative Pathnames
7. Is: Listing Directory Contents, The UNIX File System. cat: Displaying and Creating Files, cp: Copying a File, rm: Deleting Files, mv: Renaming Files
8. more: Paging Output, The lp Subsystem: Printing a File, file: Knowing the File Types, wc: Counting Lines
9. Words and Characters, od: Displaying Data in Octal, The spell and ispell, cmp: Comparing Two Files
10. Revision

Unit-2

1. ls l: Listing File Attributes, The d Option: Listing Directory Attributes, File Ownership, File Permissions
2. chmod: Changing File Permissions, Directory Permissions, Changing

Sl. No	Topic	Learning Objectives
		Unit-3
1.	SHELL Programming	Develop a command of the Unix Shell environment, including advanced Unix topics.
2	SHELL Programming	Write simple programs that manage UNIX system resources
3	SHELL Programming	Write advanced shell programs using loops and branches etc.
		Unit-4
1	Introduction to Unix System Program	Possess the skills of UNIX O.S. system programming
2	UNIX File APIs	Become acquainted with the basic tools used to develop software in the C programming language on the Unix platform.
		Unit-5
1	UNIX Processes	To introduce students the concepts and principles of UNIX Processes and to enable them to understand the duties and scope of a UNIX processes
2	Process Control	Have hands-on knowledge of the basic principles of Unix Process Control….

Review Questions

1. Explain how to create, delete, copy and rename a file in UNIX systems
2. Explain how two file can be compared in UNIX and how to display data in OCTAL
3. Explain how to check the files in current directory along with different options that can be included
4. What are Internal and external commands?
5. Explain the salient features of unix systems
6. Explain the following: gzip, gunzip and zip
7. Write note on Terminal login and Network login.
8. Explain UNIX kernel support for process.
9. Explain setjmp and longjmp function
10. Explain Memory layout of a C program
11. Explain fork and vfork

2. Understand Abstract data types, Stacks and recursion.
3. Develop and implement linked list.
4. Develop programs to implement different queues.
5. Understand and create trees.
6. Understand and implement sorting and searching techniques.

Topic Learning Objectives

After learning all the topics of unit – I, the student is able to

Sl. No	Topic	Learning Objective
Unit I		
1	Introduction to data structures	1. Define data structure(L1). 2. Write an ADT specification for rational numbers and strings(L1). 3. List (classification) the different types of data structure(L2). 4. Explain with an example classification of data structure (L2).
2	Stacks	1. Define stack (L1) 2. Define postfix and prefix expression(L1) 3. Develop an algorithm to evaluate postfix and prefix expression.(L4) 4. Develop an algorithm to convert infix to postfix and prefix to postfix(L4)
3	Recursion	1. Define Recurssion(L1) 2. Write a recursive program to find Factorial of a Number, to generate nth Fibonacci Number and Tower of Hanoi (L3) 3. List the application of stack(L2). 4. Identify the differences between recursive and iterative programs(L3)
1	Linked Lists	**UNIT-2** 1. Define Static Memory Allocation and Dynamic Memory Allocation (L1). 2. List the differences between Static Memory Allocation and Dynamic Memory Allocation (L2) 3. Identify the differences between array implementation and linked implementation (L3)

		4. Define SLL,DLL,CSLL (L1). 5. Write functions to perform basic operations on SLL,DLL,CSLL with header node and without header node (L3). 6. Develop a program to perform basic operations using above fuctions(L4)
1	Applications of Linked Lists	**UNIT-3** 1. Explain merging of two SLL, reversing a SLL, Searching an item in SLL with example(L2) 2. Develop an algorithm to merge two SLL, reversing a SLL, Searching an item in SLL (L4) 3. Write a functions to merge the given two SLL reversing a SLL, Searching an item in SLL (L3) 4. Define Polynomial(L1). 5. Given the polynomial, represent it using SLL(L2)
2	Queues	1. Define queue (L1) 2. Explain the basic operations on linear queue with an example (L2) 3. List the different methods to overcome the disadvantages of linear queue(L2) 4. Write a function for implementing queue using array and SLL(L3) 5. List the different types of queues(L2) 6. Explain the above types of queues with example(L2) 7. Write the function to implement basic operations on above queues(L3) 8. Explain the applications of queues in the field of computer science(L2)
1	Trees	**UNIT-4** 1. Define tree and the terms related to it(L1) 2. List the different tree representation(L2) 3. Define Binary tree and terms related to it(L1). 4. List types of binary trees (L2) 5. Write the algorithm for basic operations on BST (L3)
2	Trees and their applications	1. Define Tree traversal. (L2). 2. Explain different tree traversal techniques. (L1). 3. Write algorithms for different tree traversal

Text Books:

1. UNIX Concepts and Applications by Sumitabha Das, 4 edition, Tata McGraw Hill, 01-May-2006.
2. Terrence Chan: UNIX System Programming Using C++, First edition, Prentice Hall India, 2011.
3. W. Richard Stevens: Advanced Programming in the UNIX Environment, Second Edition, Pearson education, 2011

Course Outcome:

1. Understand the role of UNIX systems programming.
2. Understand UNIX System calls and terminology.
3. Able to produce programs similar to standard UNIX utilities (mv, rm etc.) using raw UNIX system calls and do basic screen manipulation (for text based editors, menu driven systems, forms etc.).
4. Writing shell programs
5. Recognize different types of file supported by UNIX operating system.
6. Knowledge of the basic principles of UNIX file system.
7. Knowledge of the basic principles of UNIX process system

Sl. No	Topic	Learning Objectives
		Unit-1
1.	Background and Basic Commands	Understand the components of UNIX OS and recognize them in different UNIX variant OS.
2	Background and Basic Commands	Understand UNIX O.S. Command line basic commands
3	The FILE System and FILE handling Commands	Understand UNIX O.S. Command line commands for file system handling
		Unit-2
1	FILE Attributes	Understand UNIX O.S. Command line commands for file handling attributes
	The Process	Understand basics of UNIX process
	Simple Filters	Understand UNIX O.S. basic input & output commands

of Command, The Logical Operators && and || -Conditional Execution, The if Conditional, Using test and [ ] to Evaluate Expressions, The case Conditional, expr: Computation and String Handling, $0: Calling a Script by Different names, while: Looping, for: Looping with a List, set and shift: Manipulating the Positional Parameters, The here Document (<<), trap: Interrupting a Program, Debugging Shell Scripts with set -x, export: Exporting Shell Variables, eval: Evaluating Twice, The exec Statement. Development of simple shell scripts to demonstrate the integer and real arithmetic operations, handling of positional parameters, the use of branching and looping constructs in the shell, handling of signals using the trap etc. 12 Hours

Unit-4

Introduction to Unix System Program: UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/ Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program 12 Hours

Unit-5

UNIX Processes: The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes.

Process Control : Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, job Control, Shell Execution of Programs, Orphaned Process Groups 12 Hours

		techniques. (L1). 3. Write algorithms for different tree traversal techniques. (L3). 4. Write binary tree for a given traversal sequences. (L3). 5. Define threaded binary tree.(L2). 6. Write the advantage of threaded binary tree (L3). 7. Write program to perform basic operations on threaded binary tree.(L3)
1	Sorting Techniques	**UNIT-5** 1. Explain different sorting techniques with example (L1) 2. Write algorithms for different sorting techniques(L4) 3. List the applications of the above sorting techniques.(L2)
2	Searching Techniques	1. Explain different searching techniques with example (L1) 2. Write algorithms for different searching techniques(L4) 3. List the applications of the above searching techniques.(L2)

Review Questions

1. Define data structure
2. Write an ADT specification for rational numbers and strings
3. List the different types of data structure
4. Explain with an example classification of data structure
5. Define stack
6. Define postfix and prefix expression
7. Develop an algorithm to evaluate postfix and prefix expression
8. Develop an algorithm to convert infix to postfix and prefix to postfix
9. Define Recursion
10. Write a recursive program to find Factorial of a Number, to generate nth Fibonacci Number and Tower of Hanoi
11. List the application of stack
12. Identify the differences between recursive and iterative programs
13. Define Static Memory Allocation and Dynamic Memory Allocation
14. List the differences between Static Memory Allocation and Dynamic Memory Allocation

15. Identify the differences between array implementation and linked implementation
16. Define SLL,DLL,CSLL
17. Write functions to perform basic operations on SLL,DLL,CSLL with header node and without header node
18. Develop a program to perform basic operations using above functions
19. Explain merging of two SLL, reversing a SLL, Searching an item in SLL with example
20. Develop an algorithm to merge two SLL, reversing a SLL, Searching an item in SLL
21. Write a functions to merge the given two SLL reversing a SLL, Searching an item in SLL
22. Define Polynomial
23. Given the polynomial, represent it using SLL
24. Define queue
25. Explain the basic operations on linear queue with an example
26. List the different methods to overcome the disadvantages of linear queue
27. Write a function for implementing queue using array and SLL
28. List the different types of queues
29. Explain the above types of queues with example
30. Write the function to implement basic operations on above queues
31. Explain the applications of queues in the field of computer science
32. Define tree and the terms related to it
33. List the different tree representation
34. Define Binary tree and terms related to it
35. List types of binary trees
36. Write the algorithm for basic operations on BST
37. Define Tree traversal.
38. Explain different tree traversal techniques.
39. write algorithms for different tree traversal techniques..
40. Write binary tree for a given traversal
41. Define threaded binary tree
42. Write the advantage of threaded binary tree
43. Write program to perform basic operations on threaded binary tree
44. Explain different sorting techniques with example
45. Write algorithms for different sorting techniques
46. List the applications of the sorting techniques
47. Write algorithms for different searching techniques
48. List the applications of the searching techniques
49. Compare the performance of different sorting techniques
50. Compare the performance of different searching techniques

Unit-2

The FILE System and FILE handling Commands

The File, The Parent-Child Relationship, The HOME Variable: The Home Directory, pwd: Checking Your Current Directory, cd: Changing the Current Directory, mkdir: Making Directories, rmdir: Removing Directories, Absolute Pathnames, Relative Pathnames, Is: Listing Directory Contents, The UNIX File System. cat: Displaying and Creating Files, cp: Copying a File, rm: Deleting Files, mv: Renaming Files, more: Paging Output, The lp Subsystem: Printing a File, file: Knowing the File Types, wc: Counting Lines, Words and Characters, od: Displaying Data in Octal, The spell and ispell, cmp: Comparing Two Files 9 Hours

Unit-3

FILE Attributes

ls l: Listing File Attributes, The d Option: Listing Directory Attributes, File Ownership, File Permissions, chmod: Changing File Permissions, Directory Permissions, Changing File Ownership. File Systems and Inodes, Hard Links, Symbolic Links and ln, The Directory, umask: Default File and Directory Permissions, Modification and Access Times, find: Locating Files, Converting One File to Other, dos2unix and unix2dos: Converting between DOS and UNIX, Compressing Files, gzip, gunzip, zip and unzip commands, tar command

The Process

Process Basics, ps: Process Status, System Processes, Mechanism of Process Creation, Internal and External Commands, Running Jobs in Background, nice: Job Execution With Low Priority, Killing Processes with Signals, Job Control, at and batch: Execute Later, cron: Running Jobs Periodically, time: Timing Processes

Simple Filters

The Sample Database, pr: Paginating Files, head: Displaying the Beginning of a File, tail: Displaying the End of a File, cut: Slitting a File Vertically, paste: Pasting Files, sort: Ordering a File, uniq: Locate Repeated and Non repeated Lines, tr: Translating Characters, An Example: Displaying a Word-count List 9 Hours

Unit-3

SHELL Programming : The Shell's Interpretive Cycle, Pattern Matching The Wild-cards, Escaping and Quoting, Redirection: The Three Standard Files, /dev/null and /dev/tty: Two Special Files, Pipes, tee: Creating a Tee, Command Substitution, Shell Variables, Environment Variables, Aliases (bash and ksh), Command History (bash and ksh). Shell Scripts, read and read-only commands, Using Command Line Arguments, exit and Exit Status

Identify and Apply algorithm Techniques to solve realistic problems.	L5	3	2	3	-	3	-	-	-	3	-	3
Design and **implement** algorithms for the given problem.	L5	3	2	3	-	3	-	-	-	3	-	3
1 – Low, 2 – Moderate and 3 – High												
Illustration of CaM of Analysis and design of Algorithms												

Course Code : P13CS45	**Semester : IV**	**L - T - P : 4-0- 0**
Course Title : Unix System Programming		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Nil		

Course Learning Objectives (CLOs)

This course aims to

1. This course will help students to achieve the following objectives:
2. Introduces the Students to the main concepts of the UNIX Operating System.
3. To familiarize with the UNIX kernel structure and system calls.
4. The most commonly used UNIX commands and utilities are described in detail as are the command line wildcard and redirection facilities.
5. Comprehensive introduction to Shell Programming.
6. To manipulate system resources such as files, processes and system information

Course Content

Unit-1

Background and Basic Commands

Brief history, Salient features of a UNIX System, The UNIX Architecture. Introduction to Linux Operating System, Internal and External Commands, Introduction to system administration, man: Browsing and Manual Pages Online, cal: The Calendar, date: Displaying and System Date, echo: Displaying a Message, printf: An Alternative to echo, bc: The Calculator, script: Recording Your Session, passwd: Changing Your Password, who, uname: Knowing Your Machine's Characteristics, tty: Knowing Your Terminal, stty: Displaying and Setting Terminal Characteristics 10 Hours

Lesson Plan

UNIT-1

1. Introduction to data structures-Definition, Abstract Data Types-ADT for rational numbers,
2. ADT for varying length Character String**,** Classification of Data Structures.
3. **Stacks** Representing stack in C
4. Implementation of Push, Pop and display operations using arrays
5. Implementation of Push, Pop and display operations using pointers
6. Example of Stacks: Infix, Postfix, Prefix
7. Infix to postfix, prefix to postfix conversion
8. Evaluation of postfix expression with example
9. **Recursion**
   Definition ,Writing Recursive programs-Factorial Numbers
10. Fibonacci Numbers and Tower of Hanoi Problem

UNIT-2

Linked lists

1. Introduction of Static Memory Allocation and Dynamic Memory Allocation
2. Explaining the different types of liked list with comparison
3. Explaining Basic operations on SLL,
4. Writing program on SLL
5. Explaining Basic operations on DLL,
6. Writing program on DLL
7. Explaining Circular SLL and Circular DLL
8. Insertion, deletion and display operations on Circular SLL
9. Insertion, deletion and display operations on Circular DLL
10. Implementation of SLL with Header nodes.

UNIT-3

1. **Applications of Linked Lists**: Explaining the application of linked list.
2. Algorithms for Merging, Reversing,
3. Addition of two polynomials using SLL
4. **Queues** Definition, Representation,
5. Implementation of queues using arrays
6. Implementation of queues using linked list
7. Different types of queues
8. Basic operations on - Linear queue, Circular queue
9. Basic operations on - Priority Queue and Double ended Queue (Using SLL).
10. Applications of Queues

UNIT-4
Trees

1. Introduction-Definition, Tree Representation.
2. Properties of Trees
3. Operations on Binary tree,,
4. Binary Search Tree [BST] - Definition
5. Searching BST, Insertion to BST,
6. Writing program , Deletion from BST, Display BST
7. Tree and their Applications- Tree Traversal, General Expression as a tree,
8. Evaluating an Expression Tree; Threaded Binary Trees-Threads
9. Inorder Traversal of a Threaded Binary Tree,
10. Inserting a Node into a Threaded Binary Tree

UNIT-5
Sorting Techniques

1. Insertion sort, Quick sort.
2. Binary tree sort
3. Explaining Heap sort
4. Writing program on Merge sort.
5. Searching Techniques sentinel search
6. Writing program probability search
7. Ordered list search
8. Comparison of different sorting techniques.
9. Comparison of different searching techniques.
10. Discussing the applications of sorting and searching techniques.

Identify and Apply algorithm Techniques to solve realistic problems.	L5	H	M	H	-	H	-	-	-	H	-	H
Design and **implement** algorithms for the given problem.	L5	H	M	H	-	H	-	-	-	H	-	H

L- Low, M- Moderate, H-High

Illustration of CAM of Analysis and design of Algorithms

Course Assessment Matrix (CaM)

Course Outcome (CO)		**Program Outcome (ABET/NBA-(3a-k))**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Analyze the space and time complexities for the given problem.	L1	3	1	2	-	-	-	-	-	-	-	-
Solve problems on searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer.	L3	3	2	2	-	3	-	-	-	3	-	3
Solve graph based problems using the different algorithm techniques.	L3	3	2	2	-	3	-	-	-	3	-	3
Apply solutions to overcome the limitations of algorithms.	L2	2	-	-	-	3	-	-	-	-	-	2

4. Branch-and-Bound for The Traveling Salesperson problem.
5. Approximation Algorithms for NP-Hard Problems.
6. Introduction to pram algorithms
7. Computational Model.
8. Parallel Algorithms for Prefix Computation.
9. List Ranking.

A. Course Articulation Matrix (CAM)

Course Outcome (CO)		**Program Outcome (ABET/NBA-(3a-k))**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Analyze the space and time complexities for the given problem.	L1	H	L	M	-	-	-	-	-	-	-	-
Solve problems on searching and sorting using the algorithm techniques such as decrease and conquer,divide and conquer.	L3	H	M	M	-	H	-	-	-	H	-	H
Solve graph based problems using the different algorithm techniques.	L3	H	M	M	-	H	-	-	-	H	-	H
Apply solutions to overcome the limitations of algorithms.	L2	M	-	-	-	**H**	-	-	-	-	-	M

Course Articulation Matrix (CAM)

Course Outcome (CO)		**Program Outcome** (ABET/NBA-(3a-k))												
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
Understand primitive and derived data structure.	L 2	-	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**
Understand Abstract data types, Stacks & recursion.	L 2	**L**	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**
Develop and implement linked list.	L 3	**L**	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**
Develop programs to implement different queues.	L 3	**L**	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**
Understand and create trees.	L 3	**L**	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**
Understand and implement sorting and searching techniques.	L 3	**M**	-	**H**	-	**M**	-	-	**M**	**H**	-	**M**	-	**H**

L- Low, M- Moderate, H-High

Course Assessment Matrix (CAM)

Course Code : P13CS34	**Semester : III**	**L - T - P : 4 - 0 - 0**
Course Title : Discrete Mathematical Structures		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Nil		

Course Learning Objectives (CLOs)

This course aims to

1. **Analyze** to solve problems using simple techniques of counting theory .
2. **Understand** the concepts of set theory extended to n case real time problems.
3. **Learn** the fundamentals of logic and its applications.
4. **Identify** Use of quantifiers, the nature of proof like direct or indirect ,proof by contradiction, check the validity of a given argument.
5. **Learn** the importance of induction principle and pigeonhole principle in proving statements.
6. **Learn** the basic concepts of Recurrence relations, Relations and functions.
7. **Identify** the different ways of representing relations in matrix and digraph form with properties.
8. **Apply** the concepts of relations and functions to solve given problem.
9. **Learn** the concepts of groups and its applications.
10. **Apply** coding theory concepts to code and encode a message

Course Content

Unit-1

Principles of counting : The rules of sum and product, Permutations, Combinations : The Binomial theorem- combinations with repetition.

Set Theory :Sets and subsets, set operations and the Laws of set theory, Counting and Venn Diagrams, A First Word on Probability . 10 Hours

Unit-2

Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence, The Laws of Logic, Logical Implication - Rules of Inference,

Quantifiers and their uses: Quantifiers, Definitions and the Proofs of Theorems (Direct and indirect methods) 10 Hours

Unit-3

Properties of Integers: Mathematical Induction, The Well Ordering Principle- Mathematical Induction in the Alternative form, Recursive definitions

Relations and Functions: Cartesian Products and Relations, Functions Plain and One-to-One, Onto Functions – Stirling's Numbers of the Second Kind, The Pigeon-hole Principle, Function Composition and Inverse Functions. Special functions-characteristic function, permutation function, Hashing function 10 Hours

5. Mathematical Analysis of non-Recursive Algorithms.
6. Mathematical Analysis of Recursive Algorithms.
7. Mathematical Analysis of Recursive Algorithms.
8. Selection sort.
9. Bubble sort.

Unit-2

1. Mergesort.
2. Quicksort.
3. Binary Search tree.
4. Binary tree traversals and related properties.
5. Defective Chess Board.
6. Insertion Sort.
7. Depth First Search.
8. Breadth First Search.
9. Topological Sorting.
10. Topological Sorting

Unit-3

1. Sorting by Counting.
2. Sorting by Counting.
3. Hors pool algorithm for Input Enhancement in String Matching.
4. Boyer-Moore algorithm for Input Enhancement in String Matching.
5. Open Hashing.
6. Closed Hashing.
7. Computing a Binomial Coefficient.
8. Computing a Binomial Coefficient.
9. Warshall's Algorithm.
10. Floyd's Algorithm.

Unit-4

1. The Knapsack Problem and Memory Functions.
2. The Knapsack Problem and Memory Functions.
3. The Knapsack Problem and Memory Functions.
4. Prim's Algorithm.
5. Kruskal's Algorithm.
6. Dijkstra's Algorithm.
7. Huffman Trees.
8. P, NP and NP-Complete Problems.
9. P, NP and NP-Complete Problems
10. Decision Trees.

Unit-5

1. Backtracking.
2. Branch-and-Bound problem.
3. Branch-and-Bound problem.

10. **Write** the Warshall's algorithm for computing the transitive closure of a directed graph

Unit-4

1. **Explain** the 0/1 knapsack problem algorithm with greedy concept.
2. **Design** a $\Theta(n^2)$ algorithm for finding the optimal BST.
3. **Write** a pseudo code for constructing a table for solving the Knapsack problem.
4. **Compute** the optimal solution to the Knapsack instance n = 7 , m = 15, and (p1, p2, p3, p4, p5, p6, p7) = (10, 5, 15, 7, 6, 18, 3) and weights (w1, w2 ,w3, w4, w5, w6, w7) = (2,3,5,7,1,4,1).
5. **Explain** the Kruskal's algorithm with an example and analyze its time complexity.
6. **Apply** Kruskal's algorithm to the given graph to find the minimum spanning tree.
7. **Apply** Prim's Algorithm to the given graph to find the minimum spanning tree.
8. **Write** and explain Diijkstra's algorithm .
9. **Explain** decision tree with example.
10. **Explain** the classes of NP-hard and NP-complete.

Unit-5

1. **Write** an algorithm of estimating the efficiency of backtracking.
2. **Draw** and explain the portion of the tree for 4-queens problem that is generated during backtracking.
3. **Explain** the applications of Backtracking.
4. **Define** the term branch and bound technique Explain it with an Example.
5. **Write** a complete LC - branch and bound algorithm for knapsack problem.
6. **Differentiate** between Dynamic Knapsack and Branch and Bound Knapsack problem.
7. **Apply** the branch and bound algorithm to solve the TSP for the following the given cost matrix.
8. **Differentiate** between NP-complete and NP-Hard.
9. **Explain** the Parallel Algorithms for Prefix Computation.
10. **Define** graph coloring? write an algorithm, which finds m-coloring of a graph.

Lesson Plan

Unit-1

1. Notion of Algorithm, Fundamentals of Algorithmic Problem Solving.Graphs.
2. Asymptotic notations and basic efficiency classes.
3. Asymptotic notations and basic efficiency classes.
4. Mathematical Analysis of non-Recursive Algorithms

Unit-4

Relations Revisited: Properties of Relations Computer Recognition : Zero-One Matrices and Directed Graphs, Partial Orders - Hasse Diagrams. Equivalence Relations and Partitions- Partitions induced by Equivalence relations. Topological sorting algorithm, Totally ordered sets . External elements , Lattices 12 Hours

Unit-5

Groups: Definitions, Elementary Properties, Homomorphisms, Isomorphisms , and Cyclic Groups, Cosets, and Lagrange's Theorem.
Coding Theory : Elements of Coding Theory, The Hamming Metric, The Parity Check, and Generator Matrices. Group Codes: Decoding with Coset Leaders 10 Hours

Text Books:

1. Discrete and Combinatorial Mathematics, Ralph P. Grimaldi,B.V. Ramana 5th Edition, PHI/Pearson Education,chapers-1, 2 , 3.1 to 3.4 ,4.1,4.2 , 5, 7.1 to 7.4, 7.6, 15.3 to 15.5, 15.7 to15.10
2. Discrete Mathematical structures –Dr D. S. Chandrashekariah .Prism 2005.

Reference Books:

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, 6th Edition, McGraw Hill, 2007.
2. Discrete Mathematical Structures: Theory and Applications, D.S.Malik and M.K. Sen, Thomson, 2004.
3. Discrete Mathematical structures 5th edition Kolman Busby Ross, PHI .

Course Outcomes :

1. **Understand** the principles of counting and set theory
2. **Identify** the quantifiers and their uses and learn the fundamentals of logic theory
3. **Apply** the Mathematical induction principle and pegion hole principle to solve the real time problems.
4. **Solve** the problems Using the concepts of relations and functions and **Identify** the different ways of representing relations
5. **Apply** the concepts of group theory & coding theory. to solve the given Problem

Topic Larning Objectives (Unitwise) (Max : 50)

Sl. No	Topic	Learning Objective
Unit I		
1	Principles of counting	1. Solve the problems of counting theory applying the rules of sum and product(L2) 2. Differentiate between Permutations and Combinations, Combinations with repetition

	Principles of counting	to apply it for the given situation.(L2) 3. Solve problems on Binomial theorem relating it to counting theory(L3) 4. Define Binomial theorem for n variables and find the coefficient of the given term in the expansion(L1) 5. Using combinations with repetition solve the real time problems in counting.(L3)
1	Set theory	1. Using laws of set theory , Membership table method (L3) 2. Venn diagram method Prove that the given two representations of sets are equal or not (L2) 3. Using addition principle for three sets and more Solve the problems of counting theory (L2) 4. Identify the laws of set theory and their uses (L2) 5. Apply Laws of set theory to represent a given set in another form.(L3) 6. Define probability and find Probability of given event (using addition principle)(L2).
1	Fundamentals of logic	**UNIT-2** 1. Define tautology, contradiction and contingency (using truth table)(L1) 2. Define Logical equivalence, Laws of logic theory, converse, inverse and contrapositive statements of a given implication .(L1) 3. Solve the problems of Logical equivalence applying the laws of logic theory 4. Find the negation of a given statement with truth valve (L3) 5. Check the validity and invalidity of the given argument expressing it symbolically (L3)
2	Quantifiers	1. Define Quantifiers, Express the statement in the symbolic form, negate the statement (L1) 2. Find the truth value of the given statement and write its negated form (L3).

3. **Define** time complexity and space complexity.
4. **Explain** the different types of asymptotic notations with examples.
5. **Show** that f(n)+g(n)=O (n2) where f(n) = 3n2- n + 4 and g(n)=nlogn+5
6. **Write** the algorithm for addition and obtain run times for n=1,10,20,30.
7. **Compute** time complexity of recursive Fibonancci procedures where F (n)= F(n-1)+ F(n-2).
8. **Write** an algorithm for matrix multiplication and find step count to calculate complexity
9. **Analyze** the computing time of selection sort?
10. **Analyze** the computing time of bubble sort?

Unit-2

1. **Write** merge sort algorithm and also find its time efficiency.
2. S**how** hoe merge sort algorithm to sort the list C, O, L, L, E, G, E in alphabetical Order.
3. **Apply** quick sort on data set 45, 50, 25, 10, 35, 25, 75, 30 .
4. **Design** a Divide and Conquer algorithm for computing the number levels in a binary tree.
5. **Justify** the equality x=n+1 by mathematical induction where x is number of external nodes and n is number of internal nodes.
6. **Design** a BFS – Based algorithm for cheking if a graph is cyclic or not.
7. **Design** a DFS – Based algorithm for cheking if a graph is cyclic or not.
8. **Distinguish Between** BFS and DFS.
9. **Define** AVL tree.
10. **Show** how heap sort sorts the following sequences of keys in ascending order 22, 55, 33, 11, 99, 77, 55, 66, 54, 21, 32.

Unit-3

1. **Design** an algorithm for the multiplying corresponding numbers from two arrays of size n, whose values are n distinct integer numbers from 1 to n.
2. **Show** how distribution sorting sorts the following sequences of keys in ascending order 45, 24, 36, 59, 68, 7, 99, 17.
3. **Explain** Horspool's algorithm for string matching with example.
4. **Apply** Horspool's algorithm to search for the pattern AT_THAT in the text WHICH_FINALLY _HALTS._ _AT_THAT.
5. **Construct** the closed hash table for the inputs 30, 20, 56,75,31,19 and the hash function h(K)=K mod 11.
6. **Distinguish Between** dynamic programming and Divide and Conquer.
7. **Compute** the fobonacci series using dynamic programming.
8. **Write** a pseudo code for computing C(n,K).
9. **Give an example** of a graph with negative weights for which Floyd's algorithm does not yield the correct result.

Sl. No	Topic	Learning Objective
		4. **Apply** Dijkstras Algorithm for finding the Single source shortest path for the given graph. 5. **Explain** the concept of Huffman Trees. 6. **Explain** the different methods for obtaining the lower bounds. 7. **Apply** the Decision Tree technique for sorting and Searching problems. 8. **Define** P,NP and NP-Complete Problems. 9. **Explain** the Decision version for any given problems. 10. **Classify** the problems according their computational complexity.
5	Copy with the Limitations of Algorithm Power ,Pram Algorithm	**Unit-5** 1. **Describe** Backtracking, Branch and Bound. 2. **Apply** the Backtracking method for solving different problems like N-Queens problem, Hamiltonian Circuit problems etc. 3. **Apply** the Branch and Bound Method for solving different problems like Assignment problem, Knapsack problem etc. 4. **Apply** the branch and bound algorithm to solve TSP for the following given cost matrix. 5. **Describe** Approximation Algorithms for NP-Hard Problems. 6. **Differentiate** between NP-complete and NP-Hard. 7. **Discover** the approximate solutions to difficult problems of combinatorial optimization using Approximation algorithms. 8. **Define** the Computational Mode. 9. **Describe** the Parallel Algorithms for Prefix Computation 10. **Define** graph coloring

Review Questions
Unit-1

1. Define an algorithm. Explain the characteristics of the algorithm
2. Explain the various stages of algorithm design and analysis process with a flow chart

		3. Check the validity of the argument (with a quantified statement)expressing it in the symbolic form.(L3) 4. Identify and find the type of proofs of theorems (Direct and indirect methods- contrapositive and contradiction methods) (L3) 5. Find the proof of the given statement in direct or indirect method.(L2)
1	Properties of Integers	**UNIT-3** 1. Define Mathematical induction principle, alternative form and Prove the given open statements truth value by Mathematical induction principle (L3) 2. Represent a sequence in two ways – recursively and explicitly, convert one form to another.(L2) 3. Disprove the given statement using MI principle(L2) 4. Use mathematical induction in the alternative form to prove statements in the recursive form.(L3) 5. Prove or disprove the given recursive statement using MI principle(L4)
2	Relations and Functions	1. Define Relations ,functions- one-one and onto functions,(L1) 2. Apply Stirling's Number of second kind to solve problems (L3) 3. Find the number of one –one functions, onto functions, bijective functions.(L1) 4. Solve the problems using pigeon hole principle (L2) 5. Define special functions-characteristic, permutation, hashing functions properties. Domain , Co domain and Range of each type of function.(L1)
1	Relations	**UNIT-4** 1. Define the properties of relations, represent the relation in the matrix form Digraph form, & Identify the relation given in any form L1 2. Prove that the given relation in an equivalence relation or partially ordered relation (L3)

2	Partially ordered relations	3. Construct the Hasse diagram for a given partially ordered relation (L3) 4. Given the Hasse diagram find the number of elements present in the set and the relation. (L3) 1. Define the properties of an equivalence relation(L1) 2. State and Prove the theorem listing the properties.(L3) 3. Find the partition induced by an equivalence relation with properties for a given relation (L2) 4. Find the external elements of a given relation - maximal, minimal, least ,greatest element ,GLB, LUB of a subset of the given set (L3) 5. Define a lattice, properties, recognize it in any form.(L1)
1 2	Group theory Coding theory	**UNIT-5** 1. Define a Group –Examples ,properties Recognize the properties of a Group, subgroup, cyclic group (L1). 2. State Lagrange’s theorem and prove it . (L2). 3. Define homomorphism, Isomorphism and cyclic groups ,cosets (L2). 4. Elementary type of groups-relations between them(L1) 1. Define elements of coding theory and Haminng metric. 2. Define Pariy check matrix,Generator matrices, coding and decoding with coset leaders.(L1) 3. Find Encode and Decode the given messages. Given the parity check matrix for a Hamming code 4. Construct a decoding table for group code given by the generator matrix.

Sl. No	Topic	Learning Objective
		5. **Define** Defective Chess Board problem. 6. **Show** the steps executed by Insertion Sort to sort the array and derive its time complexity using Decrease and conquer . 7. **Define** DFS, BFS and Topological Sorting (L1). 8. **Write** DFS, BFS traversals and Topological Sorting for the given graph using Decrease and conquer . 9. **Show** the steps executed by Heap sort to sort the array and derive its time complexity using Transform and conquer. 10. **Develop** a Devide and conquer ,Decrease and conquer algorithm, Transform and conquer algorithm to solve new problem, and derive its time complexity
3	Space and Time Tradeoffs and Dynamic Programming techniques.	**Unit-3** 1. **Explain** the concept of space and time tradeoffs. 2. **Describe** ‘Sorting by counting’ method. 3. **Illustrate** with an example for Sorting by counting method 4. **Explain** the concept of Input enhancement in string matching . 5. **Apply** Input matching string methods for different strings given 6. **Describe** Hashing and the types of Hashing. 7. **Define** B-Trees . 8. **Define** Dynamic Programming. 9. **Apply** Dynamic programming Concept to find the Binomial Coefficient. 10. **Apply** the Warshall’s Algorithm and the Floyd’s Algorithm for the graphs given .
4	Dynamic Programming , Greedy Technique , Limitations of Algorithm Power	**Unit-4** 1. **Apply** the memory function method to the Knapsack problem. 2. **Define** the Greedy Technique. **3.** **Write** Prim’s algorithm and the Kruskal’s Algorithm .**Apply** the Prims and the Kruskals algorithm for the graphs to find the minimum spanning trees

4. **Apply** branch-and-bound technique to solve assignment problem, knapsack problem and TSP
5. **Design** and implement algorithms using greedy strategy, decrease and conquer approach ,divide and conquer approach, dynamic programming approach for a given problem.
6. **Apply** the greedy strategy, dynamic programming approach, to solve problem

Topic Learning Objectives (Unit wise)

Sl. No	**Topic**	**Learning Objective**
1	Notion of Algorithm, Fundamentals of Algorithmic Problem solving, Graphs, Asymptotic notations, Mathematical Analysis of non-Recursive Algorithms and Mathematical Analysis of Recursive Algorithms.	**Unit-1** 1. **Define** Notion of Algorithm . 2. **Describe** the Fundamentals of Algorithmic Problem Solving . 3. **List** the different Algorithm Design techniques . 4. **Describe** how to prove the correctness of an algorithm . 5. **List** out the applications of graph. 6. **Explain** the Representation of graphs, digraphs and networks . 7. **Explain** the running time and space complexity of algorithms . 8. **Use** different Asymptotic notations to find the order of growth of algorithms. 9. **Use** the mathematical techniques required to prove the time complexity of program/ algorithm. (e.g., limits and sums of series). 10. **Analyze** the running time of algorithms.
2	Divide and Conquer method, Decrease and Conquer method, Transform and Conquer method .	**Unit-2** 1. **Define** divide and conquer, Decrease and conquer technique and Transform and conquer technique techniques. 2. **Describe** the three components of divide and conquer algorithms . 3. **Show** the steps executed by merge sort,quick sort to sort the array and derive its time complexity using Divide and conquer. 4. **Explain** how recurrence relations are derived from divide-and-conquer algorithms

Review Questions

1. Explain and introduce to Rule of sum and product with problems.
2. Find the number of license plates created which contains two English alphabets followed by four digits i) with repetition ii) without repetition
3. How many arrangements are there of all the letters in SOCIOLOGICAL? (i) letters A and G are adjacent? (ii) are all the Vowels adjacent?
4. Define power set ,subset , super set of A. For any three sets A,B,C Verify (A - C) - (B - C) = A- (B U C) = (A - B) – C
5. In a class of 31 students, a test of three questions was given and every student answered atleast one question, 6 students did not answer the first question, 7 failed to answer the second question and 8 did not answer the third question and 8 students answered all questions answered .Find the number of students who answered (i) exactly one question? (ii) atleast one question?
6. If two integers are selected at random and without replacement from {1,2,…,99,100} what is the probability that their sum is even.
7. If a fair coin is tossed tour times what is the probability that two heads and two tails occur.
8. Define tautology Is $(p \vee q) \rightarrow (p \rightarrow (p \wedge q))$ a tautology?(Justify your answer) using truth table and without using truth table.
9. Define logical equivalence and using laws verify $(\neg p \vee q) \wedge (p \wedge (p \wedge q)) \equiv (p \wedge q)$
10. Express symbolically and check the validity. It is not sunny this afternoon and is colderthan yesterday. We will go for swimming if and only if it is sunny. If we do not go for swimming then we will take a trip. If we take a trip then we will be home by sunset. Therefore we will be home by sunset
11. Write the statements in the symbolic form with a specific universe for each
    (i) All students have greater than 80% attendance.
    (ii) Some students have enrolled in sports
    (iii) Some integers are divisible by 5 and are even
12. Define Rule of universal specification and generalization
13. Expressing symbolically check the validity" No junior or senior has enrolled in sports. Raju has enrolled in sports. Therefore, Raju is not a senior."
14. Prove or disprove directly "The sum of any five consecutive integers is always divisible by 5".
15. State mathematical Induction principle and Prove that 1.3+2.4+……+n (n+2)= n(n+1)(2n+7)/ 6 for all integers n≥1

(i) Write the given sequence in explicit form a_1=8, and a_n=a_{n-1} + n for $n \geq 2$
(ii) Express the sequence recursively a_n=3n+2 for all $n \geq 1$

16. Define one-one functions, onto functions with example for each.
17. Find the number of one-one and onto functions from a set of m elements to a set of n elements.
18. State pegion hole principle and extended pegion hole principle.
19. Prove that any subset of size 6 from the set S={1,2,3,…,9}must contain two elements whose sum is 10.
20. Let f and g be two functions from R to R defined by f(x)=2x+1 and g(x) = x/3 Find i) fog and gof (ii) $(gof)^{-1}$ and $f^{-1}o\ g^{-1}$
21. Write the formula to find p(m ,n), S(m ,n)and p(m) . what does each number represent Counting theory
22. Define domain, codomain and range of a given function, Justify each with reason.
23. Explain Characteristic function, permutation function and hashing function and their uses
24. Let R be a relation defined as "exactly divides" on A={1, 3, 6, 9,11, 35, 385}
    a) Is R a Poset ,verify
    b) Draw the Hasse diagram of the poset .
25. Draw Hasse diagram of all positive divisors of 36
26. Define least , greatest,maximal, minimal element in a poset .
27. Let R be a relation defined as (x,y) € R iff x + y = even and S be a relation defined as
    x = y-2 on A={1,3,6,8}Find the matrix of R, S, RoS , SoR, R^2 ,S^2 .
28. Define an equivalence relation. Prove that R is an equivalence relation defined as x-y multiple of 5 on A = {0, 1, 2, 12, 15, 16}. Find the partition induced by R
29. Prove that $[M(R)]^2 = M (R^2)$ for a given relation R on A.
30. Prove that [x] = [y]or [x]∩[y]=Ø for any two elements of a poset (A,R)
31. Define Lower bound and upper bound, GLB, LUB of a subset of a poset.
32. Find GLB ,LUB of the subset of a poset.whose hasse diagram is given.
33. How to convert a partially ordered set into a totally ordered set?
34. Define a Lattice with an example.
35. Represent a Lattice in a digraph form
36. Prove that (A, "subset of") is a poset.
37. Define a Group –Examples ,properties List the properties of a Group, subgroup, cyclic group
38. State and prove Lagrange's theorem .
39. Define homomorphism, Isomorphism and cyclic groups , cosets
40. Define ,Homomorphism, Isomorphism between two groups with a n example.

Decrease and Conquer

Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting,

Transform and Conquer

Presorting, Balanced Search Trees, Heaps and Heap sort 11 Hours

Unit-3

Space and Time Tradeoffs

Sorting by Counting, Input Enhancement in String Matching, Hashing,

Dynamic Programming

Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms 10 Hours

Unit-4

Dynamic Programming

The Knapsack Problem and Memory Functions.

Greedy Technique

Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees.

Limitations of Algorithm Power

Decision Trees, P, NP and NP-Complete Problems 10 Hours

Unit-5

Copy with the Limitations of Algorithm Power

Backtracking, Branch-and-Bound, The Traveling Salesperson problem. Approximation Algorithms for NP-Hard Problems.

Pram Algorithm

Introduction, Computational Model, Parallel Algorithms for Prefix Computation, List Ranking, and Graph Problems 10 Hours

Text Books:

1. Introduction to the Design & Analysis of Algorithms**,** Anany Levitin, 2nd Edition, Pearson Education, 2007.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition, Universities Press, 2007

Reference Books:

1. Introduction to Algorithms , Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 2nd Edition, PHI, 2006.
2. Introduction to the Design and Analysis of Algorithms A Strategic Approach, R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T.Tsai, TMH, 2005.

Course Outcomes :

1. **Analyze** algorithms and find best- case, worst-case and average – case, running times of algorithms using asymptotic notations
2. **Apply** the decrease and conquer approach ,divide and conquer approach, to solve problem
3. **Describe** the notions of P, NP, NPC, and NP-hard

Course Code : P13CS44	**Semester : IV**	**L - T - P : 3-1- 0**
Course Title : Analysis and Design of Algorithms		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Subject requires student to know about

1. Basic C programming skills .
2. Elementary mathematics : Algebra – Set theory, relations.
3. Data Structure.

Course Learning Objectives (CLOs)

This course aims to

1. **Describe** the Fundamentals of Algorithmic Problem Solving .
2. **Analyze** best- case, worst-case and average - case running times of algorithms using asymptotic notations.
3. **Illustrate** the divide and conquer technique with respect to merge sort , quick sort.
4. **Illustrate** the Transform and Conquer method with respect to presorting ,heap sort .
5. **Explain** the concept of space and time tradeoffs
6. **Apply** the Warshall's Algorithm and the Floyd's Algorithm for the graphs given.
7. **Apply** the memory function method to the Knapsack problem.
8. **Define** P,NP and NP-Complete Problems.
9. **Apply** the Backtracking method for solving different problems like N-Queens
problem, Hamiltonian Circuit problems etc.
10. **Describe** the Parallel Algorithms for Prefix Computation

Course Content

Unit-1

Introduction

Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Graphs.

Fundamentals of the analysis of algorithm efficiency

Analysis Framework, Asymptotic notations and basic efficiency classes. Mathematical Analysis of non-Recursive Algorithms and Mathematical Analysis of Recursive Algorithms.

Brute Force: Selection sort, bubble sort 11 Hours

Unit-2

Divide and Conquer

Merge sort, Quick sort, Binary Search, Binary tree traversals and related properties,, Defective Chess Board.

41. Write short notes on Encoding and Decoding of a message.
42. Define Generator Matrix, Parity-check Matrix.
43. Prove that In a group code, the minimum distance between distinct code words is the minimum of the weighs of the non-zero elements of the code
44. For an encoding function $E:Z_2{}^4 \rightarrow Z_2{}^6$ is defined by the generator matrix G=

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

(i) Find the set of all code words assigned by E

(ii) Determine the associated parity check matrix

Lesson Plan

Unit-1

1. Principles of counting : Introduction to The rules of sum and product problems
2. Permutations, Combinations : Problems.
3. Explain The Binomial theorem- combinations with repetition
4. Continued - Problems
5. Set Theory : Sets and subsets, set operations and Problems
6. Addition principle of three sets and n sets ,problems.
7. Laws of set theory, Counting and Venn Diagrams, membership table Method-Problems –laws derivation.
8. Laws of logical equivalence between two given statements
9. A First Word on Probability
10. Definition- rules problems of finding Probability of the given event_All types of problems

Unit-2

1. Fundamentals of Logic: Basic Connectives and Truth Tables -problems
2. Tautology, contradiction ,contingency statements.
3. Converse, inverse, contrapositive statements Logic Equivalence-problems.
4. The Laws of Logic theory-problems.
5. Logical Implication – Argument – define, express in the symbolic form, check the validity using truth table and Rules of Inference.
6. Quantifiers and their uses: express, find the truth value, negate the given quantified statement.
7. Rule of universal specification, Rule of universal generalization-examples
8. Argument with a quantified statement, its validity and invaliity.s.

9. Problems
10. Explain different types of proof –direct indirect and contrapositive methods. Problems

Unit-3

1. Definition of Properties of Integers: Mathematical Induction, The Well Ordering Principle- Mathematical Induction in the Alternative form
2. Problems on Mathematical Induction, Mathematical Induction in the Alternative form.
3. Recursive Definitions-explicit representation
4. Relations and Functions: Cartesian Products and introduction to Relations, Functions
5. Problems of finding the domain, codomain , Range of a function. Verify whether the function is one-one, onto, or both or not
6. Stirling's Numbers of the Second Kind, The Pigeon-hole Principle
7. Problems.
8. Function Composition and Inverse Functions. Special functions
9. Problems
10. Characteristic function, Permutation function, Hashing function-Problems- Properties.

Unit-4

1. Relations Revisited: Properties of Relations –Cartesian form-Problems.
2. Computer Recognition : Zero-One Matrices , reflexive, symmetric , transitive relations
3. Problems.
4. Composition of two relations ,matrix representations of R^2 ,R^3 and so on
5. Directed Graphs, Partial Orders - Hasse Diagrams ,properties.-problems
6. Equivalence Relations- its properties, different standard relations.
7. Representation in the matrix form-its properties-theorem- proof -listing the equivalence class of every element , find the partition induced by equivalence relations.
8. Topological sorting algorithm-Problems, Extremal elements of the poset.
9. Toset and its properties, Define Lattice.
10. LUB, GLB of a subset of a POSET- Properties of a lattice.

Unit-5

1. Groups: Definitions, Elementary Properties.
2. Explain -Examples of groups subgroups, cyclic sub groups
3. Homomorphisms, Isomorphisms , and Cyclic Groups-Examples
4. Cosets, State and prove Lagrange's Theorem.
5. Introduction to coding and encoding functions
6. Elements of Coding Theory : Both to detect and correct single errors in

Course Assessment Matrix (CAM)														
Course Outcome (CO)		**Program Outcome** (ABET/NBA-(3a-k))												
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
Understand the concept of finite automata theory.	L 2	3	2	3	2	2				1			1	2
Apply regular expression for lexical analysis phases	L 3		2	3	3	3								1
Identify the syntax of higher level language	L 4		3	3	2	2							2	
Understand and classify PDA, design PDA for CFG	L 2		3	3	1	2						2	1	3
Understand Turing machine and its applications	L 2		3	3	2		3					3		2
Understand Undecidable problem, Post's Correspondence problem .	L 2		3	3	1		2					2		1
1 – Low, 2 – Moderate and 3 – High														

Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))												
		a	b	c	d	e	f	g	h	i	j	k	l	m
Understand the concept of finite automata theory.	L2	H	M	H	M	M				L			L	M
Apply regular expression for lexical analysis phases	L3		M	H	H	H								L
Identify the syntax of higher level language	L4		H	H	M	M							M	
Understand and classify PDA, design PDA for CFG	L2		H	H	L	M						M	L	H
Understand Turing machine and its applications	L2		H	H	M		H					H		M
Understand Undecidable problem, Post’s Correspondence problem .	L2		H	H	L		M					M		L

L- Low, M- Moderate, H-High

transmission.

7. The Hamming Metric,The Parity Check, and Generator Matric
8. Problems.
9. Group Codes: Decoding with Coset Leaders.
10. Problems

Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
Understand the principles of counting and set theory	L2	H	L	-	L	L	-	-	L	-	-	-
Identify the quantifiers and their uses and learn the fundamentals of logic theory	L3	H	L	-	-	L	M	H	-	-	-	-
Apply the Mathematical induction principle and pegion hole principle to solve the real time problems.	L4	H	L	-	-	L	-	-	L	-	-	-
Solve the problems Using the concepts of relations and functions and **Identify** the different ways of representing relations	L3	H	L	-	-	-	-	-	-	-	-	-
Apply the concepts of group theory and coding theory to solve the given problem.	L5	H	L	-	-	L	-	-	-	-	-	L

Course Assessment Matrix (CaM)												
Course Outcome (CO)		**Program Outcome**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Understand the principles	**L 2**	**3**	**1**	**-**	**1**	**1**	**-**	**-**	**1**	**-**	**-**	**-**
Identify the quantifiers and their uses and learn the fundamentals of logic theory	**L 3**	**3**	**1**	**-**	**-**	**1**	**2**	**3**	**-**	**-**	**-**	**-**
Apply the Mathematical induction principle and pegion hole principle to solve the real time problems.	**L 4**	**3**	**1**	**-**	**-**	**1**	**-**	**-**	**1**	**-**	**-**	**-**
Solve the problems Using the concepts of relations and functions and **Identify** the different ways of representing relations	**L 3**	**3**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
Apply the concepts of group theory and coding theory to solve the given problem.	**L 5**	**3**	**1**	**-**	**-**	**1**	**-**	**-**	**-**	**-**	**-**	**1**
1 – Low, 2 – Moderate and 3 – High												

Fig 5a. Illustration of CAM of Digital Circuits Design

3. Solving problems to convert Finite automata into Regular Expressions
4. Solving problems to convert Regular Expressions into Finite automata
5. Applications of Regular Expressions
6. Regular languages, with theorem.
7. Proving languages not to be regular languages;
8. Problems on regular languages .
9. Closure properties of regular languages;
10. Decision properties of regular languages

Unit-3

1. Context-Free Grammars And properties of Context-Free Languages: Context –free grammars; with examples.
2. Problems on CFG.
3. Explaining LMD,RMD, Parse trees.
4. Problems on LMD,RMD, Parse trees.
5. Applications of CFG;
6. Ambiguity in grammars and Language**s.**
7. Definitions of Normal forms for CFGs;
8. Problems on Normal forms of CFGs;
9. The pumping lemma for CFGs
10. Closure properties of CFLs.

Unit-4

1. Pushdown Automata: Definition of the Pushdown automata with example
2. The languages of a PDA
3. Design the PDA for the given language.
4. Explaining NPDA.
5. Problems to check whether given PDA is DPDA or NPDA.
6. Equivalence of PDA’s and CFG’s
7. Procedure to convert from PDA to CFG
8. Problems.
9. Procedure to convert from CFG to PDA
10. Problems.

Unit-5

1. Introduction to Turing Machine, Undecidability **:** Problems that Computers cannot solve;
2. The turning machine with examples.
3. Programming techniques for Turning Machines;
4. Design the TM for the given languages.
5. Extensions to the basic Turning Machines
6. Different types of TM.
7. Explaining different types of TM with examples.
8. Recursively enumerable languages

34. Design a PDA for the given CFG Design a CFG for the given PDA
35. Design a CFG for the given PDA Define DPDA with example
36. Define DPDA with example Define NPDA with example
37. Define NPDA with example What are the two conditions to be satisfied for DPDA
38. What are the two conditions to be satisfied for DPDA
39. Determine whether a given PDA is DPDA or NPDA
40. What are the advantages of PDA comparing with FA
41. Define Turing Machine
42. Design a Turing machine for the given language
43. Explain the programming techniques for Turing machine
44. List the different types of Turing machines
45. Explain the different types of Turing machines
46. Define recursively enumerable languages
47. Prove the the language is not recursively enumerable
48. List undecidable problem
49. Explain Post's Correspondence problem
50. Explain undecidable problems

Lesson Plan

Unit-1

1. **Introduction to Finite Automata** Introduction to Finite Automata ; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata. Application of finite automata; Finite automata with Epsilon transitions; Equivalence and minimization of automata.
2. Deterministic finite automata; with problems
3. Nondeterministic finite automata with problems
4. Difference between DFA and NFA by taking different examples.
5. Application of finite automata
6. Finite automata with Epsilon transitions; Difference between DFA,NFA, Epsilon NFA
7. Solving problems on Epsilon transitions;
8. Explaining the concept of Equivalence and minimization of automata.
9. Explaining the procedure to minimize the automata.
10. Solving the problems on minimization of the automata.

Unit-2

1. Regular Expression ,Regular Languages, Properties of Regular Languages Regular expressions; Definition ,regular expression for the given language.
2. Finite Automata and Regular Expressions;

Course Code : P13CS35	Semester : III	L - T - P : 4 - 0 - 0

Course Title : Object Oriented Programming with C++

Contact Period: Lecture: 52 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50

Prerequisites : Student should have programming skill in C

Course Learning Objectives (CLOs)

This course aims to

1. Understand the fundamental differences between procedure oriented and object-oriented design
2. Apply the concepts of data abstraction and data encapsulation
3. Explain the concept of redefining the operators for user defined data types
4. Understand the concept of templates to reduce code size
5. Demonstrate the ability to understand and use Exception handling and STL
6. Identify and apply the different inheritance in the given problem
7. Understand and apply multiple forms and I/O streams

Course Content

Unit-1

Basic Concepts of object oriented programming:

Objects, Classes, data abstraction and encapsulation Inheritance, polymorphism, dynamic binding, message passing. Benefits of OOP's and its application. Procedure oriented programming V/S object oriented programming (OOP)

Classes and Objects:

Creation, accessing class members, defining member functions, Inline function, function overloading, default arguments, friend function, static data members and member function, arrays of objects, object as function argument, returning objects from functions, const member function, pointer to object, namespace fundamentals 11 Hours

Unit – II

Constructor and Destructor :

Types of constructors: Parameterized constructor, multiple constructors in a class, and constructors with default arguments, copy constructor, Dynamic constructor. Dynamic initialization of objects. Destructors

Operator Overloading :

Need of operator overloading, overloading unary operators, overloading binary operators, binary operator overloading using friend function, instream / outstream operator overloading 11 Hours

Unit – III

Templates:
Introduction, function templates, function templates with multiple parameters, class templates, class templates with multiple parameters, overloading of template functions, member function templates.

Exception handling:
Exception handling fundamentals, Exception handling options.

STL: overview, containers, vectors, lists, maps 10 Hours

Unit – IV

Inheritance:
Introduction, defining a derived classes, single inheritance, multilevel inheritance, multiple inheritance, hierarchical inheritance, hybrid inheritance, Virtual base classes, constructors in derived classes 10 Hours

Unit – V

Virtual Functions and Polymorphism :
Virtual function, Calling a Virtual function through a base class reference, inheriting Virtual attribute and Virtual functions, Pure virtual functions, Early vs. late binding.

C++ I/O Stream Basics :
C++ streams, stream classes, Formatted I/O. 10 Hours

Text Books:

1. Object- oriented programming with C++,E Balguruswamy, Tata McGraw Hill, 2008.
2. Mastering C++ , K R Venugopal, RajkumarBuyya, Tata McGraw Hill, 2nd Edition, Tata McGraw Hill, 2013.

Reference Books:

1. The Complete Reference C++, Herbert Schildt, 4th Edition, Tata McGraw Hill, 2010.
2. 2. C++ Primer, Stanley B.Lippman, JoseeLajoie, 5th Edition, Pearson Education, 2005.

Review Questions

1. Define Finite automata
2. Describe the Fundamentals of automata theory
3. List the different types of Finite automata
4. Define DFA and NFA
5. Design a DFA for the given problem
6. Design a NFA for the given problem
7. Explain the applications of finite automata
8. Explain Finite automata with Epsilon transitions
9. Define equivalence of two states
10. Design the minimization of a given automata
11. Define regular expression
12. Obtain regular expression for the finite automata.
13. Explain the applications of regular expression.
14. Define Regular languages
15. Define pigeonhole principle
16. Prove that the languages are not regular Explain the closure properties of regular languages
17. Explain the closure properties of regular languages Explain the decision properties of regular languages
18. Explain the decision properties of regular languages
19. Define Context –free grammars
20. Define LMD,RMD,Parse tree
21. Write LMD,RMD and Parse tree for the given string using the given grammar
22. List the applications of grammar
23. Define ambiguous and unambiguous
24. Prove that the grammar is ambiguous
25. Obtain unambiguous grammar
26. List and define the different normal forms
27. Explain pumping lemma for CFG
28. Prove the pumping lemma for CFG
29. Define CFL
30. Explain the closure properties of CFL
31. Define PDA.
32. Define the languages of PDA
33. Design the PDA for the given languages Design a PDA for the given CFG

		7. Explain the decision properties of regular languages (L4).
3	Context-Free Grammars and properties of Context-Free Languages	**Unit-3** 1. Define Context –free grammars (L1). 2. Define LMD, RMD,Parse tree (L1). 3. Write LMD,RMD and Parse tree for the given string using the given grammar (L2). 4. List the applications of grammar (L1). 5. Define ambiguous and unambiguous (L1). 6. Prove that the grammar is ambiguous and obtain unambiguous grammar (L4). 7. List and define the different normal forms (L2). 8. Explain pumping lemma for CFG (L4). 9. Define CFL(L1). 10. Explain the closure properties of CFL (L4)
4	Pushdown Automata	**Unit-4** 1. Define PDA. (L2). 2. Define the languages of PDA (L2). 3. Design a PDA for the given CFG (L4). 4. Design a CFG for the given PDA (L4). 5. Define DPDA, NPDA (L2). **6.** Determine whether a given PDA is DPDA or NPDA (L4).
5	Introduction to Turing Machine, Undecidability	**Unit-5** 1. Define Turing Machine(L1). 2. Design a Turing machine for the given language (L4). 3. Explain the programming techniques for Turing machine (L2) 4. Explain the different types of Turing machines(L2) 5. Prove the the language is not recursively enumerable (L4). 6. Explain undecidable problem (L3). 7. Explain Post's Correspondence problem (L2**)** **8.** List undecidable problems.(L1)

Course Outcomes :

1. **Explain** the fundamental differences between procedure oriented and object-oriented design
2. **Apply** the concepts of data abstraction and data encapsulation
3. **Demonstrate** the concept of redefining the operators for user defined data types
4. **Demonstrate** the concept of redefining the operators for user defined data types
5. **Identify and apply** the different inheritance in the given problem
6. **Define and apply** multiple forms and I/O streams.

Sl. No.	Topic	Learning Objective
1	Basic Concepts of object oriented programming Classes , Objects and Namespace	1. Understand the concept of data abstraction, encapsulation, Inheritance, polymorphism, 2. Write the difference between Procedure oriented programming and Object oriented programming (OOP) 3. To identify different objects and class in a problem and define data and member functions 4. Understand the concept of Inline function, function overloading, friend function and default arguments 5. Define arrays of objects, object as function argument, returning objects from functions, 6. Understand the concept of const member function, pointer to object, namespace fundamentals.
2	Constructor and Destructor & Operator overloading	1. Understand the initialization of member variables at the time of creation and to destroy the objects 2. Write dynamic initialization of objects and dynamic constructors 3. Understand and implement Overloading the unary and binary operators for use defined data .

Sl. No.	Topic	Learning Objective
		4. Study instream /outstream operator overloading
3	Templates, Exception handling and Standard Template Libraries	1. Understand how to reduced code size by using Function templates 2. Demonstrate Function templates with multiple parameters 3. Write Class templates Class templates with multiple parameters for a given problem 4. Implement Overloading of template functions member function templates. 5. Understand the fundamentals of Exception handling and its options. 6. Demonstrate the use of standard template library to save time and to lead high quality programs
4	Inheritance	1. Understand the different types of inheritance 2. Understand to define base class and derived class 3. Implement the given problem by choosing appropriate inheritance 4. Understand and identify different Virtual base classes in a given problem 5. Understand the initialization of derived classes.
5	Polymorphism and C++ I/O Stream Basics	1. Define Virtual function and identify different virtual function in a given problem 2. Understand Inheriting Virtual attribute and Virtual functions and its application 3. Understand and implement Pure virtual functions 4. Discuss the difference between Early vs. late binding. 5. Study and use abstract concept of a file

3. Daniel I.A. Cohen: Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, 2004.
4. Thomas A. Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition, Pearson Education, 2006.

Course Outcomes:

After learning all the units of the course, the student is able to

1. Understand the concept of finite automata theory.
2. Apply regular expression for lexical analysis phases.
3. Identify the syntax of higher level language.
4. Understand and classify PDA, design PDA for CFG
5. Understand Turing machine and its applications.

Topic Learning Objectives (Unit wise)

Sl. No	Topic	Learning Objective
1	Introduction to Finite Automata, Regular Expression	**Unit-1** 1. Define Finite automata(L1). 2. Describe the Fundamentals of automata theory(L1). 3. List the different types of Finite automata (L1). 4. Define DFA and NFA (L1). 5. Design a DFA for the given problem(L4). 6. Design a NFA for the given problem(L4). 7. Explain the applications of finite automata (L2). 8. Explain Finite automata with Epsilon transitions (L2). 9. Define equivalence of two states (L1). 10. Design the minimization of a given automata (L4)
2	Regular Expression ,Regular Languages, Properties of Regular Languages	**Unit-2** 1. Define regular expression(L1). 2. Obtain regular expression for the finite automata (L3). 3. Explain the applications of regular expression (L2) 4. Define Regular languages (L1). 5. Prove that the languages are not regular (L3). 6. Explain the closure properties of regular languages (L4).

7. Design grammars from push-down automata .
8. Design push-down automata from grammars.
9. Design Turing machines for simple languages and functions.
10. Design problem reductions to determine the undecidability of languages

Course Content

Unit-1

Introduction to Finite Automata

Introduction to Finite Automata ; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata. Application of finite automata; Finite automata with Epsilon transitions; Equivalence and minimization of automata. 10 Hours

Unit-2

Regular Expression ,Regular Languages, Properties of Regular Languages

Regular expressions; Finite Automata and Regular Expressions; Applications of Regular Expressions. Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; 10 Hours

Unit-3

Context-Free Grammars And properties of Context-Free Languages

Context –free grammars; Parse trees; Applications; Ambiguity in grammars and Languages, Definitions of Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFLs. 10 Hours

Unit-4

Pushdown Automata

Definition of the Pushdown automata; The languages of a PDA; Equivalence of PDA’s and CFG’s; Deterministic Pushdown Automata. 12 Hours

Unit-5

Introduction to Turing Machine, Undecidability

Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines; Extensions to the basic Turning Machines; Turing Machine and Computers. Undecidable problem that is RE; Post’s Correspondence problem; 10 Hours

Text Book :

1. John E.. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson education, 2007.

References Books:

1. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation, Principles and Practice, Morgan Kaufmann, 1998.
2. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.

Sl. No.	Topic	Learning Objective
		providing file services and managing mass storage in C++:

Review Questions

1. How are data and functions organized in an object oriented program?
2. What is object oriented programming? How it is different from procedure oriented programming?
3. List few areas of application of OOP technology.
4. Explain the different features of OOP
5. Write a program illustrating class declaration, definition and accessing the members
6. What is the application of the scope resolution operator in C++?
7. When will you make a function inline? Why? Write inline function to find the greatest of two numbers.
8. How does an inline function differ from a preprocessor macro?
9. What are friend functions and friend classes? What are the merits and demerits of using friend functions?
10. What is this pointer? What is your reaction to the statement delete this **.**
11. What are Constructors and destructors? Explain how they differ from normal function.
12. What are the differences between default constructors and parameterized constructors?
13. List the rules to overload binary operator
14. Write a program to access members of a student class using a pointer to object members.
15. What is operator overloading? Explain the importance of the same.
16. List the limitations of overloading unary operator. How are they overcome?
17. Write a program to overload arithmetic operators for manipulating vectors.
18. The effect of a default argument can be alternatively achieved by overloading. Discuss with an example.
19. Write a function that performs the operation m^n, where m is double, n is integer. Write a function to perform the same operation but taked integer value for m. Both the functions should have the same name. Write a main that calls both the functions. Use the concept of function overloading.
20. How many arguments are required in the definition of an overloaded unary operators / Illustrate with an example
21. List the rules for overloading Operator
22. Define template. Explain the use of writing a template

23. What is function template? Write a function template for finding the largest numbers in a given array. The array parameter must have generic data types.
24. Distinguish between the terms class template and template class.
25. A class (or function) template is known as a parameterized class(or function). Comment
26. What is an exception? How is an exception handled in C++?
27. What should be used placed in try and catch block?
28. What is STL and list its components? How STL algorithms are different from the conventional algorithms?
29. Distinguish between lists and vectors, sets and maps.
30. Suggest a suitable containers for the following application i) insertion at the back of a container. Ii) Frequent insertion and deletion at both the ends of a container. iii) Frequent insertion and deletion in the middle of a container.
31. What does inheritance mean in C++?
32. What are the differences between inheriting a class with public and private visibility mode?
33. Describe the syntax of different types of inheritance.
34. What are the implications of the following two definitions?
    Class A: public B, public C{//….};
    Class A: public C, public B{//….};
35. What is a virtual base class? When do we make a class virtual?
36. Explain how base class member functions can be invoked in a derived class if the derived class also has a member function with the same name.
37. What are virtual classes? Explain the need for virtual classes while building a class hierarchy.
38. Consider an example of declaring the examination result. Design threeclasses: **Student**, **Exam** and **Result**. The Student class has data members such as those representing roll no., name, etc. Create the class exam by inheriting the Student class. The Exam class adds data members representing the marks scored in six subjects. Derive the Result from the Exam class and it has its own members such as total marks. Write an interactive program to model this relationship. What type of inheritance does this model belongs to?
39. Explain the general form of defining a derived constructors

3. **Apply** optimization techniques to construct a minimal spanning tree of a graph, Prefix code for a given message.	**L4**	**3**	**1**	**-**	**-**	**2**	**-**	**1**	**1**	**-**	**-**	**3**
4. **Apply** and Understand the principle of inclusion and exclusion, generating functions to solve the given problem.	**L5**	**3**	**1**	**-**	**-**	**-**	**-**	**1**	**-**	**-**	**-**	**1**
5**. solve** simple recurrence relationof second and third order .	**L5**	**3**	**1**	**-**	**-**	**1**	**-**	**1**	**-**	**-**	**-**	**2**

1 – Low, 2 – Moderate and 3 – High

Fig 5a. Illustration of CAM of Digital Circuits Design

Course Code : P13CS43	**Semester : IV**	**L - T - P : 4 - 0 - 0**
Course Title : Theory of Computation		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Subject requires student to know about

1. Basic C programming skills
2. Elementary mathematics : Algebra – Set theory, relations.
3. Discrete mathematics
4. Data Structure

Course Learning Objectives (CLOs)

This course aims to

1. Design finite automata.
2. Explain equivalence and minimization of finite automata.
3. Design regular expression for regular languages, convert between finite automata and regular expressions for regular languages.
4. Apply the pumping lemma for regular languages to determine if a language is regular.
5. Design grammars for various languages
6. Demonstrate that grammar is ambiguous.

A. Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
1..**Identify** different paramenters of graphs and its applications	L 2	H	L	-	-	L	-	-	-	-	-	L
2.**Understand** planar graphs and its propertiesTo detect planarity of a given graph	L 3	H	L	-	-	L	-	L	-	-	-	L
3.**Apply** optimization techniques to construct a minimal spanning tree of a graph, Prefix code for a given message.	L 4	H	L	-	-	M	-	L	L	-	-	H
4. **Apply** and Understand the principle of inclusion and exclusion, generating functions to solve the given problem.	L 3	H	L	-	-	-	-	L	-	-	-	L
5. **Solve** simple recurrence relation of second and third order .	L 5	H	L	-	-	L	-	L	-	-	-	M

L- Low, M- Moderate, H-High

Fig 3. Illustration of CAM of Digital Circuits Design

A.
Course Assessment Matrix (CaM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
1.**Identify** different paramenters of graphs and its applications	L2	3	1	-	-	1	-	-	-	-	-	1
2.**Understand** planar graphs and its properties to detect planarity of a given graph	L3	3	1	-	-	1	-	1	-	-	-	1

40. Describe how an object of a class that contains objects of other classes created?
41. What does polymorphism mean in C++ language? How it is achieved at compile time and run time?
42. Why do we need virtual functions? When do we make a virtual function “pure”? What are the implications of making a function a pure virtual function?
43. What are rules that must be satisfied while creating virtual functions?
44. Differentiate between early and late binding by giving an example to each.
45. What are streams? Explain the features of C++ stream I/O with C’s I/O system.
46. What is the output of the following statements
 cout<<65 ; ii) cout.put(65); iii) cout.out(‘A’);
47. Explain the various methods of performing formatted stream I/O operations.
48. Why are the words such as cin and cout not considered as keywords?
49. What is the role of file() function? When do we use this function?
50. Discuss the various forms of get() function supported by the input strem. How are they used.

Lesson Plan

1. Basic Concepts of object oriented programming: Objects, Classes, data abstraction and encapsulation Inheritance, polymorphism, dynamic binding, message passing. Benefits of OOP’s and its application. Procedure oriented programming V/S object oriented programming (OOP), Classes and Objects: Creation, accessing class members.
2. Defining member functions, Inline function, function overloading, default arguments, friend function, static data members and member function.
3. Arrays of objects, object as function argument, returning objects from functions, const member function, pointer to object, namespace fundamentals, Introduction to constructor and destructor
4. Types of constructors: Parameterized constructor, multiple constructors in a class, and constructors with default arguments, copy constructor
5. Dynamic constructor. Dynamic initialization of objects. Destructors. Need of operator overloading, rules to overload operators. overloading unary operators, overloading binary operators,
6. Binary operator overloading using friend function, instream / outstream operator overloading , Introduction, function templates

7. Function templates with multiple parameters, class templates, class templates with multiple parameters, overloading of template functions, member function templates.
8. Exception handling fundamentals, Exception handling options.: overview, containers, vectors, lists, maps.
9. Introduction, defining a derived classes and base class, Protected data member , single inheritance. Example for single inheritance
10. Multilevel inheritance, Multiple inheritance , hierarchical inheritance ,Hybrid, inheritance with examples
11. Examples contd. , Virtual base classes, constructors in derived classes. Introduction to Virtual function
12. Calling a Virtual function through a base class reference, inheriting Virtual attribute and Virtual functions, Pure virtual functions, Early vs. late binding.
13. C++ streams, stream classes, Formatted I/O.

Course Articulation Matrix (CAM)												
Course Outcome (CO)		**Program Outcome (ABET/NBA-(3a-k))**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**I**	**j**	**k**
Explain the fundamental differences between procedure oriented and object-oriented design	**L 2**			H								
Apply the concepts of data abstraction and data encapsulation	**L 3**		M			H						
Demonstrate the concept of redefining the operators for user defined data types	**L 3**		H			H						H
Illustrate the concept of templates to reduce code	**L 4**		M	M		M						H

UNIT-3

1. Trees: Definitions, properties, and examples
2. Theorems proving the propertiesand their proofs.
3. Rooted trees, trees and sorting
4. Applications-expalination of applications of trees
5. Weighted trees and prefix codes.-Problems.
6. Optimization: Dijkstra's shortest path algorithm, -
7. minimal spanning trees-Fundamental cutsets and circuits.
8. The algorithms of Kruskal and Prim to find minimal spanning tree.
9. Finding minimal spanning tree by both algorithms.
10. Transport networks - Maxflow,Min-cut theorem –problems .

UNIT-4

1. The principle of inclusion and exclusion –Introduction with derivation.
2. The principle of inclusion and exclusion Generalizations of the principle.
3. Derangements, Nothing is in its right place
4. Review of the problems.
5. Rook polynomials.-Problems.
6. Generating functions: Introductory examples,–Problems.
7. Continued for problems.
8. Calculational techniques-problems.
9. Partitions of integers, The exponential generating function,
10. The summation operator-problems.

UNIT-5

1. Recurrence relations with constant coefficients.
2. First order linear recurrence relation-problems.
3. Solving Problems.
4. The second order linear homogeneous recurrence relation.
5. Solving problems
6. Third and higher –order Homogeous Recurennce relations,
7. Solving problems.(continued)
8. The non homogeneous recurrence relation **–**Problems.
9. The method of generating functions for second order recurrence relations. Problems.

41. In how many ways can 12 oranges be distributed among three children A, B ,C so that A gets at least lour B, and C get at least two but C gets no more than five?
42. Explain Exponential generating function.
43. Solve the recurrence relation $a_{n+1} = 4a_n$ for $n \geq 0$ given that $a_0 = 3$
44. Find the recurrence relation and the intial condition for the sequence 0,2,6,12,20,30, 42……..
45. A bank pays a certain % of annual interest on deposits , compounding the interest once in 3 months. If a deposit in 6 years and 6 months. What is the annual %of interest paid by the bank ?
46. Explain Second –order Homogeneous Recurrence Relations
47. Solve the recurrence relation $a_n = 3a_{n-1} - 2a_{n-2}$ for $n \geq 2$ given that $a_1 = 5$ and $a_2 = 3$
48. Solve the recurrence relation $2a_{n+3} = a_{n+2} + 2a_{n+1} - a_n$ for $n \geq 0$ with $a_0=0, a_1=1, a_2=2$

Lesson Plan

UNIT-1

1. Introduction to Graph Theory : Definitions and examples, complements.
2. Different types of graphs ,sub graphs, Operations on graphs,
3. Graph isomorphism. -Problems
4. Vertex degree, Euler Trails,Hamiltonian circuits
5. Theorem proofs-to recognize the existence of graphs.
6. Application of Graphs-Introduction.Problems.
7. Konigsberg Bridge problem, Travelling salesmen problem ,
8. Properties of standard graphs.
9. Utility problem, Seating arrangement problem.
10. Problems.

UNIT-2

1. Planar graphs, Introduction
2. Kuratowski’s two graphs-proofs of the theorems.
3. different representations of a planar graphs,
4. Eulers formula-theorem statement and proof
5. Detection of planarity. Geometric dual , Geometric dual .
6. Coloring : Cutsets , some properties of a cut-set, ,
7. Graph colouring of all types of graphs.
8. Discussion of all type of graphs and chromatic number.
9. Chromatic partitioning and chromatic polynomials.
10. Problems.

Course Articulation Matrix (CAM)												
Course Outcome (CO)		Program Outcome										
		a	b	c	d	e	f	g	h	I	j	k
Demonstrate the ability to understand and use Exception handling and STL	**L 3**		H	H		M						*H*
Identify and apply the different inheritance in	**L 1**		H	H								*H*
Define and apply multiple forms and I/O streams.	**L 1 & L**		M	H		M						*H*
L- Low, M- Moderate, H-High												

Fig 3. Illustration of CAM of Digital Circuits Design

Course Code : P13CS36	**Semester : III**	**L - T - P : 4 - 0 - 0**
Course Title : Computer Organization		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Subject requires student to know about
1. Logic circuits 2. Basics of computers

Course Learning Objectives (CLOs)

This course aims to

1. Understand the basic operational concepts,bus structures.
2. Understand instruction sequencing, addressing modes, Basics of assembly language, number representation.
3. Understand the concept of accessing I/O devices, Interrupts, DMA, Excep tions.
4. Explain different types of memories with their functionalities.
5. Understand the design and working of fast adders.
6. Understand different algorithms for performing arithmetic operations.
7. Explain the concept of bus organization, pipelining and multiprocesssors.
8. Understand the different types in generation of control signals.

Course Content

Unit-1

Basic structure of computers: Computer types, Functional units, Basic operational concepts, Bus structures, Performance.

Machine instructions & programs: Numbers, arithmetic operations & characters, Memory location & addresses, Memory operations, Instructions & instruction sequencing; Addressing modes, Assembly language, Basic input/ output operations, Stacks & queues, Subroutines, Additional instructions, Encoding of machine instructions 11 Hours

Unit-2

Input/output Organization: Accessing I/O devices, Interrupts-Interrupt hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct memory access, Buses, Interface circuits, Standard I/O Interfaces. 10 Hours

Unit-3

Memory system: Basic concepts, Semiconductor RAM memories, Read-Only memories, Cache memories-Mapping Functions, Replacement Algorithms, Performance considerations, Introduction to Virtual memory. 10 Hrs

Unit-4

Arithmetic: Addition & subtraction of signed numbers, Design of fast adders; Multiplication of positive numbers, Signed-operand multiplication,

13. Write the steps involved in drawing the dual of the given planar graph .
14. State and prove Euler's formula.
15. Define proper coloring of a graph, chromatic number and Find the chromatic number of a Peterson's graph.
16. Prove that a tree is always 2- chromatic.
17. State Decomposition theorem.
18. Find the chromatic polynomial of a given graph.
19. Find the chromatic number of a tree , bipartite graph ,complete.
20. Define Tree ,rooted tree ,Weighed tree, m-ary tree, Binary tree, Balanced tree.
21. Prove that Every tree with n vertices has n-1 edges.
22. The computer laboratory of a school has 10 computers that are to be connected to a wall socket that has 2 outlets. Conncetions are made by using extension cords that have 2 out lets each. Find the least number of cords needed to get these computers set up for use.
23. How many internal vertices does complete 5 ary tree with 817 leaves have?
24. How many leaves does a complete 6-ary tree of order 733 have?.
25. Using merge sort method sort the list 7,3,8,4,5,10,6,2,9.
26. Explain the difference between DFS and BFS spanning trees of a graph with an example.
27. Write the steps involved in Prim's and Kruskal's algorithm.
28. Obtain an optimal prefix code for the message MISSION SUCCESSFUL . Indicate the code for the message
29. Using the Dijkstra's algorithm find the shortest path and its weigh from the vertex 1 to each of the other vertices in the given directed graph
30. State Max flow and mincut theorem.
31. Find the number of nonnegative integer solutions of the equation x_1+ x_2 + x_3+ x_4 = 18 under the conditions $x_i \le 7$ for i=1,2,3,4.
32. Define Derangements and Find the number of derangements of 1,2,3,4
33. In how many ways can the integers 1 to 10 be arragemd in a line so that no even integer is in its natural place.
34. Explain Rook polynomial.
35. Find the rook polynomial for a 2x2 board by using the expansion fomula.
36. Obtain the formula for d_n the number of derangements of n objects by using rook polymonials .
37. Find the sequences generated by the following functions $(3+x)^3$, $(1+3x)^{-1/3}$
38. Find a generating function for the following sequences
39. (i) 1,1,0,1,1,1,… (ii)0,2,6,12,20,30,42,….
40. Determine the coefficient of x^5 in the expansion of $(1-2x)^{-7}$

		4. **Find** the number of partitions for a given positive integer using generating functions. (L3) 5. **Find** the exponential generating function for the given sequence.(L3)
1	Recursive relations	**Unit-5** 1. **Explain** First order Linear recurrence relation with constant coefficient.(L2) 2. **Obtain** the recurrence relation and initial condition for the given sequence.(L2) 3. **Explain** second order linear homogeneous recurrence relation with constant coefficient.(L2) **4.** **Solve** the given second order recurrence relations given.(L3)
2	Third and higher order	1. **Explain** Third and higher order Homogeous and non homogeneous recurrence relation. 2. **Obtain** the recurrence relation and initial condition for the given sequence.(L2) 3. **Solve** the second orderrecurrence relation using method of generating functions

Review Questions

1. Define complete graph, regular graphs ,finite ,connected graphs. with an example for each
2. Prove that a complete graph with n vertices and e edges will have n(n-1)\2 edges.
3. P.T. $\Sigma\ d(v_i)$=2 e. for any graph
4. Define isomorphism(i)draw two graphs that are isomorphic. ii) draw two graphs that are not isomorphic but have same number of vertices and edges.
5. Explain the applications of graph theory with usual notations
6. Write short notes on Konignsberg bridge problem, travelling salesman problem, seating arrangement problems.
7. Construct a graph that is complete , regular, connected .
8. Differentiate between Euler graphs and Hamiltonian graphs and their uses.
9. Construct a graph that is both Euler and Hamiltonian and list its properties
10. Define planar graphs and prove that k5 and k3,3 are non planar graphs
11. State Kuratowski's theorem
12. Detect planarity of a given graph applying kuratowski's theorem.

Fast multiplication, Integer division, Floating point numbers and operations.
10 Hours

Unit-5

Basic processing unit: Some fundamental concepts, Execution of a complete instruction, Multiple bus organization, Hardwired control; Micro programmed control, Basic concepts of pipelining,
The structure of general purpose multiprocessors, memory organization in multiprocessors , 11 Hours

Text Book:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5th Edition, TMH, 2002.

Reference Book:

1. Computer Organization & Architecture, William Stallings, 7th Edition, PHI, 2006.
2. Computer Systems Design and Architecture, Vincent P. Heuring & Harry F. Jordan, 2nd Ed. Pearson Education, 2004.

Course Outcomes :

1. Understand and analyze the machine instructions and program execution.
2. Understand and Explain the I/O organization
3. Understand and explain the memory system.
4. Apply the algorithms used for performing various arithmetic operations.
5. Understand the operation of pipelining and multiprocessor

Topic Learning Objective

Unit-1

Basic structure of computers.

Machine instructions & programs.

1. **Understand** the basic structure of a computer.
2. **Write** machine instructions and understand their execution including branching, and subroutine call and return operation.
3. **Analyse** System software that enables the preparation and execution of programs
4. **Explain** Number representation and addition/subtraction in the 2's complement system.
6. **Identify** different Addressing methods for accessing register and memory operands.
7. **Understand** how Program control input / output operations are performed.
8. **Explain** different Operations on stack, queue.
9. **List** out and perform different Shift operation types.
10. **Understand** the concept of Encoding-1, 2, 3 word instructions.
11. **Understand** the concept of Subroutines

Unit-2

Input/output Organization

1. **Understand** how Program controlled I/O is performed using polling.
2. **Understand** the idea of interrupts and the hardware and the software needed to support them.
3. **Explain** Direct memory access I/O mechanism for high speed devices.
4. **Differentiate** between data transfer over synchronous and Asynchronous buses.
5. **Understand** the design of I/O interface circuits.
6. **Identify** Commercial bus standards in particular PCI, SCSI, USB buses.

Unit-3

Memory system

1. **Understand** the concept of Basic memory circuits.
2. **Explain** Organization of the main memory
3. **Understand** Cache memory concept, which shortens the effective memory access time.
4. **Understand** Virtual memory mechanism, which increases the apparent size of the main memory**.**

Unit-4

Arithmetic

1. **Design** High speed adders implemented in a hierarchical structure using carry look ahead
2. logic to generate carry signals in parallel.
3. **Apply** the Booth algorithm to determine how multiplicand summands are selected by the
   multiplier bit patterns in performing multiplication of signed numbers.
4. **Design** Circuits that perform division operations.
5. **Explain** the representation of floating point numbers in IEEE standard format and how to
   perform basic arithmetic operations on them.

Unit-5

Basic processing unit

1. **Analyze** how a processor executes instructions
2. **Understand** the internal functional units of a processor and how they are inter connected.
3. **Design** Hardware for generating internal control signals.
4. **Explain** the microprogramming approach .
5. **Understand** Micro program organization.

Review Questions

1. Explain the functional units of a computer
2. Describe how the performance of the computer is measured?
3. Explain how parameters are passed to the subroutine. Write a program to multiply list of 'n' numbers stored in the memory, which calls the subroutine LISTMUL and trace the same with suitable example.

Unit 3		
1	Trees	1. **Explain** tree, forest, spanning tree, rooted tree, directed tree, binary tree.(L2) 2. **Construct** a rooted tree for the given expression and to find the expression in Polish notation.(L3) 3. **Apply** the preorder, postorder and in order traversal techniques on a rooted tree.(L3) 4. **Apply** BFS and DFS methods to find the minimal spanning tree.(L3) 5. **Construct** optimal prefix codes for the given symbols with the given frequencies.(L3)
2	Optimization	1. **Explain** Dijkstras algorithm.(L1) 2. **Apply** Dijkstras algorithm to find the shortest path from single source to all other vertices.(L3) 3. **Explain and Apply** Prims and Kruskals algorithm to find the minimum spanning tree for the given graph. (L3) 4. **Find** the maximum flow and corresponding min-cut for the given transport network using max-flow Min –cut theorem.(L3)
1	Principles of Inclusion and Exclusion	**Unit-4** 1. **Apply** the principles of inclusion and exclusion, to determine the number of positive integers that satisfy the given condition.(L3) 2. **Explain and List** derangements.(L2) 3. **Find** the number of derangements for the given number.(L2) 4. **Explain** Rook polynomial.(L2) 5. **Find** the rook polynomial for the given chess board.(L3)
2	Generating functions	1. **Define** Generating functions.(L1) 2. **Find** the generating functions for the given sequence.(L3) 3. **Explain** different techniques for finding the generating function.(L3)

A. Topic Learning Objectives (Unitwise) (Max : 50)

Sl. No	Topic	Learning Objective
		Unit I
1	Introduction to graph theory	1. **Define** basic terminologies of graph (L1). 2. **Apply** the basic properties of graph like to find the walk, trial, circuit…etc.(L3) 3. **Apply** definition of graph isomorphism to check if the two graphs are isomorphic or not.(L3) 4. **Construct** the graphs whose properties are given (L2)
2	Applications	1. **Explain** Hamilton cycle, path.(L2) 2. **Determine** a given graph has Hamilton path or cycle.(L3) 3. **Explain** Euler graphs with examples(L3) 4. **Identify** the different type of problems which lead to know the applications of graph theory (L2) 5. **Explain** Konigsberg bridge problem, travelling salesman problem, utility problem etc.
		Unit 2
1	Planar graphs	1. **Explain** main Planar graph, Bipartite graph, graph homomorphism(L2) 2. **Apply** Kuratowski's theorem to check the planarity of the graphs (L3). 3. **Derive** Eulers formula.(L3) 4. **Detect** the planarity of a graph(L3) 5. **Construct** the dual of a given graph(L2)
2	Coloring	1. **Explain** graph coloring problem, chromatic number, chromatic polynomial(L2) 2. **Find** the chromatic number and polynomial and partition of standard graphs.(L3) 3. **Determine** the chromatic number and polynomial for a given graph using Decomposition and multiplication theorem(L3)

4. Explain with example all generic addressing modes with assembler syntax.
5. What is the function of assembler directives? Give 2 examples of assembler directives used for reservation of memory locations for variables. state their functions.
6. What is word alignment of a machine? Explain.What are the consecutive addresses of the aligned words 16, 32, 64 bits word length of machine? Give 2 consecutive addresses for each.
7. Explain the important technological features and devices that characterized each generation of computers.
8. Explain shift and rotate operations with example.
9. Define subroutine. Explain subroutine linkage using a link register.
10. In modern computers why interrupts are required? Support your claim with suitable example.
11. In the interrupt mechanism, how simultaneous arrivals of interrupts from various devices are handled.
12. Define bus arbitration. List and explain various approaches to Bus arbitration.
13. Explain, with the help of a diagram the working of daisy chain with multiple priority levels and multiple devices in each level.
14. Define exceptions. Explain 2 kinds of exceptions.
15. Explain the following: (i) Interrupt concepts (ii) Interrupt hardware.
16. Define cycle stealing, Burst mode.
17. With a neat sketch explain the individual input and output interface circuits. Also list their salient features.
18. In the computer system why PCI bus is used.
19. Explain with a block diagram a general 8 bit parallel interface.
20. Define and explain the following: (i) Memory access time (ii) Memory Cycle time (iii) Random access memory (iv) Static memory.
21. Differentiate between the static RAM and Dynamic RAM giving 4 key differences. State the primary usage of SRAM and DRAM in contemporary computer systems.
22. Explain a simple method of translating virtual address of a program into the physical, with the help of a diagram.
23. Draw a neat block diagram of memory hierarchy in a contemporary computer system. Also indicate the relative variation of size, speed and cost/ bit in the hierarchy
24. With the block diagram explain the operation of a 16 bit megabit DRAM configured as 2MX8.
25. Explain different mapping functions used in the cache memory

.

26. What is memory interleaving? Explain.
27. Explain the concept of carry save addition for the multiplication operation, MXQ=P for 4-bit operands, with diagram and suitable example.
28. In a carry look ahead adder explain the generate Gi and Propagate Pi function for stagei with the help of Boolean expression for Gi and Pi.
29. Perform signed multiplication of numbers -12 and -11 using booth multiplication algorithm. Represent the numbers in 5-bit including the sign bit. Give booth multiplier recoding table that is used in the above multiplication.
30. Perform the division of 8 by 3 using non-restoring division algorithm.
31. Explain how a 16-bit carry look ahead adder can be built from 4 bit adder.
32. Show the multiplication of (+13) and (-6) using multiplier bit pair recoding technique.
33. Differentiate between restoring and non-restoring division algorithms.
34. Explain the IEEE standard for floating point number representation.
35. Explain the process of fetching the word from the memory using timing diagram of memory read operation. Also give an example for the same.
36. Write the control sequence of execution of the instruction ADD (R3),R1 using single bus organization.
37. Write and explain the control sequence for the execution of an unconditional branch instruction.
38. Explain with block diagram the basic organization of a microprgrammed control unit.
39. What are the modifications required in the basic organization of a microprogrammed control unit to support conditional branching in the micrprogram.
40. Explain with diagram how control signals are generated using single bus organization.
41. Explain the multibus organization.
42. Draw the block diagram of a complete processor and identify the units.

Lesson Plan

Week-1

Basic structure of computers: Computer types, Functional units, Basic operational concepts, Bus structures, Performance, Numbers, arithmetic operations & characters,

Unit-3

Trees: Definitions, properties, and examples, rooted trees, trees and sorting, Weighted trees and prefix codes.

Optimization: Dijkstra's shortest path algorithm, minimal spanning trees - The algorithms of Kruskal and Prim, Transport networks - Maxflow,Min-cut theorem 10 Hours

Unit-4

The principle of inclusion and exclusion: The principle of inclusion and exclusion, Generalizations of the principle, derangements, Nothing is in its right place, Rook polynomials.

Generating functions: Introductory examples, Definition and examples–calculational techniques, partitions of integers, The exponential generating function, The summation operator 12 Hours

Unit-5

Recurrence relations: First order linear recurrence relation, the second order linear homogeneous recurrence relation with constant coefficients.

Third and higher –order Homogeous Recurennce relations,The non homogeneous recurrence relation, The method of generating functions for second order recurrence relations. 10 Hours

Text Books :

1. Discrete and Combinatorial Mathematics, Ralph P.Grimaldi B.V.Ramana ,5th Edition, PHI/Pearson education. Chapters 8,9,10,11,12.
2. Graph Theory with Applications to Engineering and Computer Science - Narsing Deo. Chapters-1,2,3,4.1,4.2,58.1 to 8.4.

Reference Books :

1. Graph Theory and Combinatorics, Dr. D.S. Chandrasekharaiah, Prism, 2005.
2. Introduction to Graph Theory, Chartrand Zhang, TMH, 2006.

Course Outcomes

After learning all the units of the course, the student is able to

1. **Identify** different paramenters of graphs and its applications
2. **Understand** planar graphs and its properties To detect planarity of a given graph
3. **Apply** optimization techniques to construct a minimal spanning tree of a graph, Prefix code for a given message
4. **Apply** and Understand the principle of inclusion and exclusion, generating functions to solve the given problem.
5. **Solve** simple récurrence relation of second and third order .

Course Code : P13CS42	**Semester : IV**	**L - T - P : 2-1- 0**
Course Title : Graph Theory & Combinatorics		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	

Prerequisites : Subject requires student to know about
Elementary mathematics: Algebra – Set theory, relations.
Data Structure & Basics of Computer science

Course Learning Objectives (CLOs)

This course aims to

1. **Develop** the ability to identify different types of graphs, vertex degree, Euler trial, Euler circuit to find the Hamilton path, cycle, finding the chromatic polynomial for a given graph.
2. **Explain** Directed tree, rooted tree, binary rooted tree and the applications of rooted trees. and Construct optimal tree for the given prefix codes.
3. **Apply** Dijkstra's algorithm to find the shortest path from single source to all other vertices. Prim's and the Kruskal's algorithm to construct the minimal spanning trees.
4. **Apply** the principles of inclusion and exclusion theorem, generalization principle for the given problem**.**
5. **Apply** the concept of generating functions; find the number of partitions of a positive integer for the given generating function.
6. **Learn** methods for solving simple recurrence relations of second and third order.

Course Content

Unit-1

Introduction to Graph Theory : Definitions and examples, finite and infinite graphs ,sub graphs, Operations on graphs, complements, and Graph isomorphism.

Applications : Vertex degree, Euler Trails and circuits ,complements, Hamilton paths and cycles. Application of Graphs-Konigsberg Bridge problem, Travelling salesmen problem , Utility problem, Seating arrangement problem**.** 10 Hours

Unit-2

Planar graphs, Kuratowski's two graphs, different representations of a planar graphs, Eulers formula, Detection of planarity. Geometric dual , Geometric dual Coloring : Cutsets , some properties of a cut-set Graph colouring , chromatic number, chromatic partitioning and chromatic polynomials 10 Hours

Week-2
Memory location & addresses, Memory operations, Instructions & instruction sequencing; Addressing modes,
Week-3
Assembly language, Basic input/output operations, Stacks & queues, Subroutines, Additional instructions, Encoding of machine instructions, **Input/ output Organization:** Accessing I/O devices,
Week-4
Interrupts-Interrupt hardware, Enabling and Disabling Interrupts, Handling Multiple Devices
Week-5
Controlling Device Requests, Exceptions, Direct memory access,
Week-6
Buses, Interface circuits, Standard I/O Interfaces. **Memory system:** Basic concepts,
Week-7
Semiconductor RAM memories, Read-Only memories, memories-Mapping Functions,
Week-8
Replacement Algorithms, Performance considerations, Introduction to Virtual memory.
Week-9
Arithmetic: Addition & subtraction of signed numbers, Design of fast adders; Multiplication of positive numbers,
Week-10
Signed-operand multiplication, Fast multiplication, Integer division, Floating point numbers and operations.
Week-11
Basic processing unit: Some fundamental concepts, Execution of a complete instruction
Week-12
Multiple bus organization, Hardwired control; Micro programmed control,
Week-13
Basic concepts of pipelining, The structure of general purpose multiprocessors, memory organization in multiprocessors.

Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome (ABET/NBA-(3a-k))										
		a	b	c	d	e	f	g	h	i	j	k
Understand and analyze the machine instructions and program execution.	**L2, L4**	M	M	M	-	-	-	M	-	L	-	M
Understand and Explain the I/O organization	**L2**	-	M	M	-	-	-	M	-	L	-	-
Understand and explain the memory system.	**L2**	M	M	H	-	M	-	-	-	M	-	M
Apply the algorithms used for performing various arithmetic operations.	**L3**	M	H	M	-	H	-	-	-	-	-	-
Understand the operation of pipelining and multiprocessor.	**L4**	H	H	M	-	M	-	-	-	-	-	-

L- Low, M- Moderate, H-High

Fig 3. Illustration of CAM of Digital Circuits Design

Decoder and Arithmetic circuits

1. 1 bit and 2 bit Magnitude comparator using Decoder IC and NAND gates.
2. Full adder and Full subtractor using Decoder IC and NAND gates.
3. Write the Verilog/VHDL code for full subtractor and Full adder.

Encoder

1. Implement an 8-to-3-line encoder
2. Write the Verilog/VHDL code for encoder.

Experiment on Flip-Flops

1. JK Master slave using Nand gates.
2. Convert JK flip flop to D and T flip flop
3. Write the Verilog/VHDL code for D Flip-Flop with positive-edge triggering.

Shift Register

1. Implement a ring counter and Johnson counter using 4-bit shift register.
2. Design a 3-bit serial-in –serial out and a parallel-in –parallel out shift register using J-K flip flop
3. Design a 3 bit sequence detector and verify its operation
4. Write the Verilog/VHDL code for ring and Johnson counter.

Counters

1. Implement an asynchronous counter using decade counter and use 7 segment display to display the count 0 to n (n<=9).
2. Write the Verilog/VHDL code for decade counter.
3. Design and implement 3 bit (n<8) synchronous up counter using J-K Flip -Flop ICs.
4. Design a counter for the given sequence with lock in condition.
5. Design a 2 bit down counter using D- Flip –Flop.
6. Write the Verilog/VHDL code for sequence counter

iii) To count the number of nodes in a tree iv) To display the contents

18. Write a C program to search en element in a given list of ‘n’ numbers using i) Linear Search ii) Binary Search

Course Code : P13CSL38	**Semester : III**	**L - T - P : 0:0.5:1**
Course Title : Digital Logic Design Laboratory		
No. of Hours per Week: 3, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Student should have programming skill in C and knowledge of Data Structures		

Course Outcomes:

Students will be able to :

1. Develop and implement programs on stacks
2. Develop and implement programs on recursion
3. Develop and implement programs on queues
4. Develop and implement programs on linked lists
5. Develop and implement programs on trees.

Course Content

Experiment on combinational logic circuits

1. Introduction to basic gates.
2. Realization of boolean expressions.

Code Converters

1. Binary to Grey code using basic gates.
2. Excess -3 to BCD using universal gates and display the result in 7 segment display.
3. Write the Verilog /VHDL code for both (a) and (b).

Experiment on data processing circuit.

1. Given any 3 and 4 variable logic expression simplify using EVM and realize the simplified logic expression using
   - * 3:8 multiplexer.
   - * Use suitable Decoder
2. Write the Verilog /VHDL code for an 8:1 multiplexer.

Course Assessment Matrix (CaM)												
Course Outcome (CO)		**Program Outcome**										
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**
Understand and analyze the machine instructions	**L2, L4**	2	2	2	-	-	-	2	-	1	-	2
Understand and Explain the I/O organization	**L2**	-	2	2	-	-	-	2	-	3	-	-
Understand and explain	**L2**	2	2	3	-	2	-	-	-	2	-	2
Apply the algorithms used for performing various arithmetic operations.	**L3**	2	3	2	-	3	-	-	-	-	-	-
Understand the operation of pipelining and	**L4**	3	3	2	-	2	-	-	-	-	-	-
1 – Low, 2 – Moderate and 3 – High												

Fig 5a. Illustration of CAM of Digital Circuits Design

Course Code : P13CSL37	**Semester : III**	**L - T - P : 0:0.5:1**
Course Title : Data Structures Laboratory		
No. of Hours per Week: 3, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Student should have programming skill in C and knowledge of Data Structures		

Course Outcomes:

Students will be able to :

1. Develop and implement programs on stacks
2. Develop and implement programs on recursion
3. Develop and implement programs on queues
4. Develop and implement programs on linked lists
5. Develop and implement programs on trees.

Course Content

1. Write a C program to construct a stack and to perform the following operations.
   i) Push ii) Pop iii) Display
   The program should print appropriate message for stack overflow, stack underflow & stack empty.
2. Write a C program to convert and print a given valid parenthesized infix arithmetic expression to prefix expression. The expression consists of single character operands and binary operators + (Plus), - (Minus), * (Multiply), / (Divide).
3. Write a C program to evaluate a valid prefix expression using stack. Assume that the prefix expression is read as single line consisting of non negative single digit operands and binary arithmetic operations.
4. Write a C program to check whether a given string is palindrome or not using stack.

Programs on Recursion

5. Write a recursive C programs for
   a) To find larger of 'n' elements in an array
   b) To multiply two natural numbers
   c) Solving the Towers of Hanoi Problem

Programs on Queues

6. Write a C program to simulate the working of a queues using an array provide the following operation
   i) Insert ii) Delete iii) Display
7. Write a C program to simulate the working of a circular queues with items as strings. Provide the following operations
   i) Insert ii) Delete iii) Display
8. Write a C program to simulate the working of Double Ended Queue of integers using Structures. Provide the following operations
   i) Insert from front/rear end ii) Delete from front/rear end iii) Display
9. Write a C program to implement priority queues using structures (Assume a maximum of 3 queues).

Programs on Linked List

10. Write a C program using dynamic variables and pointers, to construct a Singly Linked List consisting of the following information in each node : Employee id (integer), Employee name (character string) and Department (character string). The operation to be supported are
    a) The insertion operation
    i) At the front end of the list ii) At the rear end of the list
    iii) At any portion in the list

- Deleting a node based on employee id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.
- Searching a node based on employee id and updates the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.
- Displaying all the nodes in the list

11. Write a C program to construct a **Ordered Singly Linked List** and to perform the following operations
    i) Reverse a list ii) Concatenation of two lists

- Write a C program to support the following operations on a Doubly Linked List where each node consists of integers
- Create a Doubly Linked List by adding each node at the front
- Insert a new node to the right of the node whose key value is read as an input
- To delete all nodes whose info is same as key item.
- Display the contents of the list

Programs on Tree

12. Write a C program
i) To create a tree ii) To search for an item
iii) To get the exact copy of a tree iv) To display the elements

13. Write a C program
To construct a binary search tree of integers
To traverse the tree using In-Order, Pre-Order and Post-Order traversal method
To display the elements

14. Write a C program
To construct a ordered BST of items
To insert an item into an ordered BST (No duplicates are allowed)
To search an item in BST
To display the elements

15. Write a C program to sort the given list of 'n' numbers using
i) Merge Sortii) Quick Sort

Exercise Problems

16. Write a C program to implement queues using Singly Linked List.
17. Write a C program
i) To create a binary tree ii) To find the height of a tree