Syllabus

Out Come Based Education

III & IV Semester
Bachelor Degree
in
Information Science and 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. (Information Science and Engineering)

Sl No	Course Code	Course Title	Teaching Dept.	Hours /Week L:T:P:H	Credit	Examination Marks		
						CIE	SEE	Total Marks
1.	P13MAT31	Course I - Engineering Mathematics-III	Maths	4:0:0:4	4	50	50	100
2.	P13IS32	Digital Design	ISE	2:2:0:4	3	50	50	100
3.	P13IS33	Data Structures	ISE	4:0:0:4	4	50	50	100
4.	P13IS34	Discrete Mathematical Structures	ISE	4:0:0:4	4	50	50	100
5.	P13IS35	Computer Organization	ISE	4:0:0:4	4	50	50	100
6.	P13IS36	Object Oriented Programming with Java	ISE	3:0:2:5	4	50	50	100
7.	P13ISL37	Data Structures Lab	ISE	0:0:3:3	1.5	50	50	100
8.	P13ISL38	Digital Design Lab	ISE	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	P13ISL310	Industry Interaction - I	ISE	0:0:1:1	0	(50)	--	--
11	P13HM311	Constitution of India & Professional Ethics	Human& Science	2:0:0:2	0	(50)	---	---
12	P13MADIP31	Additional Maths-I *	Maths	4:0:0:4	0	(50)	---	---
13	P13HUDIP39	English & Persona Evolution#	HS&M	4:0:0:4	[2]#	[50]#	[50]#	[100]#
		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.
ACPA- **I** All students shall have to pass this mandatory learning courses before completion of V - Semester.
#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 : P13ISL47	**Semester : IV**	**L - T - P : 1.5- 0 - 0**
Course Title : Microprocessor Laboratory		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites :		

Course Learning Objectives (CLOs)

This course aims to
1. Implement programs related to Searching , sorting and strings
2. Implement programs using Macro , procedure and files
3. Implement Data conversion , number generation.
4. Interface hardware device to 8086 processor

Course Content

Part A

1. Program on searching and sorting,
2. Macros and Procedures(subroutines)
3. Data conversions
4. Strings
5. Recursion
6. File(Read and write)
7. Number Generation

Part B
(Using Interfacing concepts)

1. Programs on logic controller,
2. Programs on7-segment display,
3. Programs on stepper motor interface,
4. Programs on 8X3 Keypad,
5. Programs on elevator interface.

Note: Out of the exercises executed in the regular laboratory classes, each students will be allotted one question from Part A and one question from Part B, by taking lots in the SEE ,which he has to execute individually.

Course Code : P13ISL47	Semester : IV	L - T - P : 1.5 - 0 - 0
Course Title : Algorithms Laboratory		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50	
Prerequisites :		

Course Learning Objectives (CLOs)

This course aims to

1. Distinguish between the basic concepts of time and space complexity and various design strategies
2. Apply the methodologies of Brute force and Divide and conquer and evaluate the complexity.
3. Solve a problem using Transform and conquer algorithms and evaluate its correctness.
4. Formulate the time-complexity analysis for Dynamic programming and greedy techniques.
5. Apply, Analyze and Design Branch and Bound techniques

Course Content

PART-A

NOTE: Design and Implement programs for the following Algorithms using RAPTOR TOOL, C or C++ Languages: .

1. Programs to implement Brute Force method.
2. Programs to implement Divide and Conquer method.
3. Programs to implement Decrease and conquer method.
4. Programs to implement Space and Time Tradeoffs.
5. Program to implement Dynamic Programming Method.
6. Program to implement Greedy method.

P.E.S. COLLEGE OF ENGINEERING, MANDYA
(An Autonomous Institution)
SCHEME OF TEACHING AND EXAMINATION
IV Semester B.E. (Information Science and Engineering)

Sl No.	Course Code	Course Title	Teaching Dept.	Hours/ Week L:T:P:H	Credit	Examination Marks		
						CIE	SEE	Total Marks
1	P13MAAC41+/ P13MAES41++	Course I - Engineering Mathematics-IV (HC)/	Maths	4:0:0:4	4	50	50	100
2	P13IS42	Analysis and Design of Algorithms	ISE	2:2:0:4	3	50	50	100
3	P13IS43	Finite automata and formal languages	ISE	4:0:0:4	4	50	50	100
4	P13IS44	Operating System	ISE	4:0:0:4	4	50	50	100
5	P13IS45	Graph Theory and Combinatorics	ISE	4:0:0:4	4	50	50	100
6	P13IS46	Microprocessor	ISE	4:0:0:4	4	50	50	100
7	P13ISL47	Analysis and Design of Algorithms Lab	ISE	0:0:3:3	1.5	50	50	100
8	P13ISL48	Microprocessor Lab	ISE	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	P13ISL410	Mini Project- I	ISE	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			26	400	400	800

L: Lecture, T: Tutorial, P: Practical's, H: Hours/Week, CIE: Continuous Internal Evaluation, SEE: Semester End Examination
* Additional Mathematics-I & Constitution of India and Professional Ethics : Lateral entry students shall have to pass these mandatory learning courses before completion of V- 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 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)	L1	M	-	-	-	M	-	-	-	-	-	-
Apply 8086 instruction set for the given problems (Unit-II)	L3	H	H	L	-	-	M	-	-	-	-	-
Develop different modules & link them. (Unit-III)	L6	H	H	L	-	M	-	-	-	-	-	-
Apply string instruction set and I/0 Interrupt in 8086 programming (Unit-IV)	L3	H	H	L	-	M	-	-	-	-	-	-
Understand min & max mode of 8086. (Unit-V)	L1	M	-	-	-	-	-	-	-	-	-	-

L- Low, M- Moderate, H-High

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)	**L1**	2	-	-	-	2	-	-	-	-	-	-
Apply 8086 instruction set for the given problems (Unit-II)	**L3**	3	3	1	-	-	2	-	-	-	-	-
Develop different modules & link them. (Unit-III)	**L6**	3	3	1	-	2	-	-	-	-	-	-
Apply string instruction set and I/0 Interrupt in 8086 program-ming (Unit-IV)	**L3**	3	3	1	-	2	-	-	-	-	-	-
Understand min & max mode of 8086. (Unit-V)	**L1**	2	-	-	-	-	-	-	-	-	-	-

1 – Low, 2 – Moderate and 3 – High

Unit – III

1. Introduction to Modular Programming
2. Linking and Relocation
3. Segment Combination
4. Access to External Identifiers
5. Stacks
6. Procedures – Calls, Returns and Procedure Definitions
7. Saving and Restoring Register
8. Interrupts and Interrupt Routines
9. MSAM Macros.
10. Examples
11. Review

Unit – IV

1. Introduction to Byte and String Manipulation
2. String Instructions
3. REP prefix
4. table translation
5. I/O programming: Fundamental I/O considerations,
6. Programmed I/O,
7. Interrupt I/O,
8. Block transfers and DMA.
9. Examples
10. Review

Unit – V

1. Introduction to System Bus Structure
2. Basic 8086/8088 configurations
3. Minimum mode
4. Maximum mode,
5. System Bus Timing
6. Interrupt Priority Management – Interrupt System based on Single 8259A,
7. Interrupt System Based on Multiple 8259As
8. Bus Standards.
9. Examples
10. Review

Course Code : P13IS32	**Semester : III**	**L - T - P : 2 - 1 - 0**
Course Title : Digital Design		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Maths1 and Maths2		

Course Learning Objectives (CLOs)

This course aims to

1. Explain how digital circuit of large complexity can be built in a methodological way, starting from Boolean logic and applying a set of rigorous techniques.
2. Provide numerous examples and case studies will be used to illustrate how the concepts presented in the lectures are applied in practice.
3. Make students to apply their knowledge in the labs by building increasingly more complex digital logic circuits.
4. Provide a bridge to future courses where alternative designs and their performance trade-offs will be studied in depth.
5. Appreciate certain aspects of the physical realization of digital circuits.
6. Make students familiar with basic sequential logic components: SR Latch, D Flip-Flop and their usage and able to analyze sequential logic circuits.
7. Understand basic combinational and sequential components used in the typical data-path designs: Register, Adders, Shifters, Comparators; Counters, Multiplier, Arithmetic-Logic Units (ALUs), RAM.
8. Convert digital circuit to analog circuit and vice-versa.
9. Make the students familiar with one of the de-facto standard HDLs, (VHDL) and gain experience in programmable logic devices.
10. Write VHDL code for logic circuits.

Course Content

Unit – I

Boolean Algebra and Combinational Networks & Simplification of Boolean Expressions:

Definition of a Boolean Algebra-Boolean Theorems and Functions-Canonical Formulas- Manipulations of Boolean Formulas-Gates and Combinational Networks-Incomplete Boolean Functions and Don't Care Conditions-Additional Boolean operations and Gates Formulation of the simplification problem. Prime Implicants And Irredundant Disjunctive Expressions-Prime Implicates and Irredundant Conjunctive Expressions and 2, 3 and 4 variable K-Map and The Quine-Mccluskey Method-To find Prime Implicants using decimal method and binary method upto 5 variables.

10 Hrs

Unit – II

Arithmetic Circuits & Data processing Circuits:

Binary Adders & Subtracters, Decimal Adders, Code converter full Adder and Full Subtractor. Magnitude Comparators, Multiplexers and Demultiplexers, Decoders, Parity generator and checkers, Encoders.

10 Hrs

Unit – III

Memory and Programmable logic ,DAC Converters , VHDL Language:

Programmable Logic Devices- RAM, ROM, PROMs, PLA's, PAL. D/A Conversion and A/D Conversion: Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, VHDL – Introduction to VHDL, describing data flow, behavioural, structural and mixed design style, Simulating design for arithmetic and combinational circuits.

11 Hrs

Unit – IV

Flip-Flops and Registers:

Clock Waveform – Characteristics of ideal Clock Waveforms, synchronous operation, propagational delay time. The Basic Bistable Element-Latches-Timing Considerations-JK Master –Slave Flip-Flops, Pulse-Triggered Flip flops, Edge-Triggered Flip-Flops-Characteristic Equations. Conversions of Flip Flop. Types of Registers, Serial In – Serial Out, Serial In – Parallel out, Parallel In – Serial Out, Parallel In – Parallel Out using JK or D Flip Flops. Applications of Shift Registers. Ring counter, Johnson counter, sequence detector and sequence generator.

11 Hrs

Unit – V

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.

Design of Synchronous and Asynchronous Sequential Circuits:

Design of Synchronous Sequential Circuits- Model Selection, State Transition Diagram, State Synthesis Table, Design Equations and Circuit Diagram, State Reduction Technique. Analysis of Asynchronous Sequential Circuit, Problems with Asynchronous Sequential Circuits, Design of Asynchronous Sequential Circuit.

VHDL code for counters.

10 Hrs

Text Book:

1. "Digital Principles & Design", Donald D Givone, 4th Reprint, Tata McGraw Hill 2009.
2. "Digital Principles and Applications", Donald P Leach, Albert Paul Malvino and Goutham Saha, TMH, 7th Edition, 2006.

Reference Books:

1. "Fundamentals of Digital Logic Design with VHDL", Stephen Brown, Zvonko Vranesic, 2nd Edition, Tata McGraw Hill, 2005.
2. "Digital Systems Principles and Applications", Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, 10th Edition, Pearson Education, 2007.
3. "Digital Logic and Computer Design", M Morris Mano, 10 Edition,Pearson Education, 2008.

34. Explain the usefulness of the following instructions in 8086.
35. Can we write following instructions for microprocessor 8086? a.MOV CX AL c.MOV DS,437AH c.MOV CL,[BX],DL d.MOV43H[SI],DH
36. Explain theruse of the following prefixes.
a.REP b.REPE
37. Discuss all types of jump instructions used in 8086 microprocessor.
38. Write a operations performed by the 8086 microprocessor CALL instruction.
39. What is the function of assembler directives
40. Explain variables, suffix and operators used in assembly language programming.
41. What do you mean by machine language program?
42.What do you mean by assembly language program?
43.Give the difference between machine language,assembly language and high level language.
44. Explain the steps that assembler follows to convert .ASM file to .OBJ file
45.Explain the function of linker.
46. What is debugger ? Explain its advantages.
47. Write an assembly program to sort ten 8-bit numbers.
48. Write an assembly program to find out odd and even numbers of an array. Store the result in separate arrays.
49. What is stack.
50. What is the function of stack pointer.

Lesson Plan

Unit – I

1. Introduction
2. **8086 Architecture:** CPU architecture
3. Internal Operation
4. Machine Language Instructions
5. addressing modes
6. Instruction formats
7. Instruction Execution
8. Timing
9. Examples
10. Review

Unit – II

1. Introduction to Assembler language programming**,** Assembler instruction format
2. Data transfer Instructions, arithmetic instructions
3. Branch instructions- conditional branch instruction, unconditional branch instructions
4. Loop instructions, NOP and HLT instructions
5. Logical Instructions, Shift and Rotate Instructions
6. Directives and operators- data definition and storage allocation
7. Structure, records, assigning name of expression
8. Segment definition, program termination
9. Alignment directives
10. value returning Attribute Operators
11. Examples

Review Questions

1. Draw and explain the block diagram of simple microprocessor based system.
2. Write a short note on terminologies used in microprocessor.
3. Explain fetching, decoding and execution operations of microprocessor.
4. What is stack? What do you mean by stack pointer?
5. What is the function of CPU in microcomputer system?
6. Explain the software hierarchy.
7. Explain the features of 8085.
8. Give the clock out frequency and state time, T,of an 8085A operating with each of the following frequency crystals :6.25MHz,5MHz and 4MHz.
9. What do you mean by hardware interrupts?
10. Explain how 8085 response to INTR interrupt.
11. What do you mean by masking the interrupt ? How is it achieved in 8085?
12. What do you mean by pending interrupts?
13. Define 1.Instructioncycle 2.Machinecycle 3.T state.
14. Explain various machine cycles supported by 8085
15. Draw and explain the I/O read cycle of 8085.
16. Draw and explain the I/O write cycle of 8085.
17. Explain various clock circuits used in 8085 system.
18. What is the use of ALE signal?
19. Why buffers are required in the microprocessor system?
20. Write a short note on bus drivers.
21. Explain the register organisation of 8086.
22. What are segment registers? Explain the purpose of them.
23. Explain the purpose of pointers and index registers.
24. What is the function of flag register?
25. Explain the architecture of 8086 processor with the help of neat block diagram.
26. Draw a bit pattern for flag register of 8086 and explain the significance of each bit.
27. List the rules for memory segmentation
28. What is the function of SI and DI registers?
29. With the help of blick schematic diagrams explain the operation of 8282 clock generator and 8286 transceiver.
30. Define buscycle, and explain the minimum mode read and write bus cycle with proper timing diagram.
31. Explain the HOLD response sequence in the minimum mode of 8086 with the help of timing diagram.
32. Draw and explain a block diagram showing 8086 in maximum mode configuration.
33. Indicate the signals which are different when 8086 in minimum mode and in maximum mode.

Course Outcomes

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

1. Solve basic binary math operations using the logic gates.
2. Demonstrate programming proficiency using the various logical elements to design practically motivated logical units.
3. Design different units that are elements of typical computer's CPU.
4. Apply knowledge of the logic design course to solve problems of designing of control units of different input/output devices.
5. Explain the operation of basic building blocks of a flip-flop that is constructed of gates and latches.
6. Analyse and design Asynchronous and Synchronous Sequential circuits.

Topic Learning Objectives

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

1. Simplify Boolean algebraic expressions by applying Boolean algebra theorems.– L3
2. Express functions in canonical sum-of-product and canonical product-of-sum forms.– L2
3. Determine the function performed by a combinational- or sequential-logic circuit through analysis.– L4
4. Design the logic circuit using basic gates.-L3
5. Construct the logic circuit using universal gates.-L3
6. Minimize the cost, complexity, power, and latency of a combinational- or sequential-logic circuit by applying various logic minimization approaches, including Boolean simplification.– L5
7. Know the importance of prime implicants and to write Irredundant Disjunctive Expressions.– L2
8. Minimize the cost and complexity of a combinational- or sequential-logic circuit by applying minimization approach Karnaugh – maps method.– L5
9. Minimize the cost and complexity of a combinational- or sequential-logic circuit by applying minimization approach Quine-Mccluskey Method.– L5
10. Find Prime Implicants using decimal method and binary method upto 5 variables.– L3

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

1. Define combinational circuits and to be able to build simple applications using encoders/decoders, (de) multiplexers, exclusive-ORs, comparators. – L1
2. Design decimal adders.– L3
3. Design Code converters.– L3
4. Design Full adder and Full subtractor.– L3
5. Explain how a Magnitude Comparator works.– L2
6. Design combinational-logic building blocks including multiplexers.– L3
7. Design combinational-logic building blocks including Demultiplexers.– L3
8. Construct a circuit for decoder.– L2
9. Explain the purpose of parity checking.– L2
10. Construct a circuit for encoder.– L2

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

1. Explain the need of Programmable Logic Devices. – L2
2. Discuss PLA and PAL. – L2
3. Design A/D Converter. – L3
4. Define Binary ladder.-L1
5. Explain the use of Binary Ladders with an example. – L2
6. Design D/A Converter. – L3
7. Explain the need of VHDL. – L2
8. Write the VHDL code for logic circuit.-L3
9. Discuss data flow, behavioural, structural and mixed design style. – L2
10. Compare dataflow and behavioural models.-L3

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

1. State the purpose of a clock in a digital system and demonstrate an understanding of basic terms and concepts related to clock waveforms. – L3
2. Define propagation delay. – L1
3. Explain the operation of a Bistable element. – L2
4. Discuss some of the timing problems related to flip-flop. – L2
5. Draw a diagram of a JK master-slave flip-flop and describe its operation. – L2
6. Derive the characteristic equation of various flip-flops. – L3
7. Convert SR flip-flop to JK flip-flop. – L3
8. Demonstrate different types of registers using JK Flip flop. – L3
9. Discuss the applications of Shift registers. – L2
10. Explain Ring counter, Johnson counter and sequence detector. – L2

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

1. Design Mod-6 Asynchronous up counter. – L3
2. Design Mod-7 Asynchronous down counter. – L3
3. Design Mod-6 Synchronous up counter. – L3
4. Design Mod-7 Synchronous down counter. – L3
5. Explain the operation of a decade counter with an example. – L3
6. Explain the steps involved in designing a synchronous Sequential Circuit. – L2
7. Explain the steps involved in designing a Asynchronous Sequential Circuit. – L2
8. Write a VHDL code for Mod-8 up counter. – L3
9. Write a VHDL code for Mod-8 down counter. – L3
10. Compare synchronous and Asynchronous counter.-L3

Unit – V

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 hrs**

Text Books

1. The Intel Microprocessors, 7th Edition, Barry B. Brey, Pearson / PHI 2006

Reference Books

1. Microprocessor and Interfacing , Douglas V.Hall, Revised 2nd Edition , TMH , 2006.
2. Advanced Microprocessors and IBM-PC assembly Language Programming, K. Udaya Kumar and B.S. Umashankar, TMH 2003

Course Outcomes

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

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

Topic Learning Objectives

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

1. Understand the introduction about general Microprocessor. L2
2.Explain 8086 CPU architecture. L2
3.Analyze the internal operation of 8086 Microprocessor. L4
4.Determine Machine Language Instructions L4
5.Apply instruction formats. L3
6.Discuss Instruction Execution Timing L6

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

1.Understand Assembler instruction format L2
2.Analyze Data transfer Instructions L4
3.Discuss arithmetic instructions L6
4.Determine branch & loop instructions L4
5.Explain logical Instructions L2
6.Discuss about directives and operators L6

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

1.Understand the concept of Linking & Relocation of Modular Programming L2
2.Describe Access to External Identifier L2
3.Understand about Stacks L2
4.Understand about Procedures L2
5.Explains Interrupts and Interrupt Routines L2
6.Explains MSAM Macros L2

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

1.Describe String Instructions and Solve it L2
2.Explains Fundamental I/O considerations L2
3.Understand Programmed I/O L2
4.Explains Interrupt I/O L2
5.Understand Block transfers and DMA L2

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

1.Explains basic 8086/8088 configurations L2
2.Describe System Bus Timing L2
3.Discuss Interrupt Priority Management L6
4.Understand Bus Standards L2

Course Code : P13IS46	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 :		

Course Learning Objectives (CLOs)

This course aims to

1. Understand the architecture of 8086 microprocessor
2. Apply 8086 instruction set for given problems
3. Develop different modules and link them
4. learn how the hardware and software components of a microprocessor-based system work together to implement system-level features;
5. learn both hardware and software aspects of integrating digital devices (such as memory and I/O interfaces) into microprocessor-based systems;
6. understand different modes (min and max) of 8086
7. apply string instructions and I/O interrupts in 8086 programming
8. learn the operating principles of, and gain hands-on experience with, common microprocessor peripherals
9. get practical experience in assembly-language programming;
10. get exposed to the tools and techniques used by practicing engineers to design, implement, and debug microprocessor-based systems (during the Lab).
11. Identify, formulate and solve engineering problems in the microprocessor based system design considering the following

Course Content

Unit – I

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

10hrs

Unit – II

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 hrs**

Unit – III

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. **11hrs**

Unit – IV

Byte and String Manipulation: String Instructions, REP prefix, table translation. **I/O programming:** Fundamental I/O considerations, Programmed I/O, Interrupt I/O, Block transfers and DMA.

10 hrs

Review Questions

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. Simplify the given expression using Boolean laws. Write a logic circuit for simplified expression;

$$Y= \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\ \overline{D} + \overline{A}\overline{B}\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D}$$

3. Implement F = (CD+E) (A+B') using NAND gates.
4. Implement the following Boolean function with NAND – NAND logic and NOR – NOR logic.

$$Y= AC + ABC + \overline{A}\overline{B}\overline{C} + AB + D$$

5. 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
6. Convert the given expression in standard SOP form;
   F(A,B,C) = A+AB+CB
7. Convert the given expression in standard POS form;
   F(P,Q,R)=(P+Q') (P+R)
8. Represent each of the following Boolean functions on a Karnaugh map:
   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')
9. 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)
10. 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)
11. Explain the procedure for loading a K-map using map entered variable technique with an example.
12. Realize a full adder using minimum number of two input NAND gates. Write the relevant expressions, truth table and logic diagram.
13. Draw and explain the block diagram of n-bit parallel adder.
14. Realize a full subtractor using basic gates only.
15. What is a high speed adder? Design a 4 bit carry look ahead adder circuit.
16. Design
    i. BCD to Excess – 3 code converter
    ii. Binary to gray code converter
17. Design and explain two bit magnitude comparator.
18. Discuss the implementation of the following function using 4:1 multiplexer
    F (a,b,c,d) = ∑m(0,1,2,4,6,9,12,14)
19. Implement the following function using 8:1 multiplexer:
    F(w,x,y,z) = ∑m(0,1,5,6,8,10,12,15)

20. What is decoder? Using gates, show how do you design a 3-to-8 line decoder.
21. Explain a decimal to binary encoder using four OR gates. What is a priority encoder?
22. What is PLA? How PLA differs from PAL.
23. Design the logic circuit for odd parity checker.
24. Define the following terms for D/A converters:
    i. Resolution ii. Accuracy iii. Monotonicity iv. Conversion time
25. Obtain an expression for the output voltage of R/2R DAC.
26. Define VHDL. Write the VHDL code 8:1 multiplexer.
27. Write the VHDL code for full adder and full subtractor.
28. Write the VHDL code for the following expressions;
    $f(a,b,c)= a\bar{b}\bar{c} + abc + a\bar{b}$
29. Define clock cycle time. Explain with neat waveform.
30. State and explain the characteristics of ideal clock waveform.
31. Explain the working of S-R flip-flop by using NOR gates only.
32. Define race around condition? Explain how it is eliminated.
33. Explain the operation of the master- slave JK flip-flop along with a circuit diagram.
34. Derive the characteristic equation of various flip-flops.
35. Convert SR flip-flop to JK flip-flop
36. Convert JK flip-flop to T flip-flop
37. Explain the working of S-R flip-flop by using NAND gates only.
38. List the applications of Shift Registers.
39. How are shift registers used for sequence generator & sequence detector?
40. Explain the operation of 4 bit serial in serial out and parallel in serial out shift register with the help of a neat circuit diagram for the 4 - bit data 1101
41. Design 3 -bit Johnson counter and ring counter.
42. Design 4-bit sequence detector and generator.
43. Design mod-3 counter using JK flip-flop. Sketch the waveforms for outputs when clock is applied and verify its operations.
44. Design a synchronous counter to count the sequence 4→2→1 using D-Flip Flop.
45. Explain with the help of neat diagram 4 – bit asynchronous decade counter.
46. Design an excess -3 decimal counter using JK flip-flops.
47. Differentiate between synchronous and asynchronous counter.
48. Design Mod-7 synchronous up counter.
49. Design Mod-5 asynchronous down counter.
50. Write the VHDL code for Mod-5 up counter.

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
1..**Identify** different paramenters of graphs and its applications	H	M	**L**									M	
2.**Understand** planar graphs and its propertiesTo detect planarity of a given graph	**H**	**M**					**L**					M	
3.**Apply** optimization techniques to construct a minimal spanning tree of a graph, Prefix code for a given message.	**H**	**M**	**M**	**L**								M	
4. **Apply** and Understand the principle of inclusion and exclusion, generating functions to solve the given problem.	**H**	**L**	**L**	**M**								M	

L- Low, M- Moderate, H-High

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

1 – Low, 2 – Moderate and 3 – High

1. Continued for problems.
2. Calculational techniques-problems.
3. Partitions of integers
4. The exponential generating function
5. The summation operator-problems
6. Problems

Unit – V

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

Lesson Plan

Unit – I

1. Definition of a Boolean Algebra-Boolean Theorems
2. Functions-Canonical Formulas, examples
3. Manipulations of Boolean Formulas-Gates
4. Combinational Networks-Incomplete Boolean Functions
5. Don't Care Conditions-Additional Boolean operations
6. Gates Formulation of the simplification problem
7. Prime Implicants and Irredundant Disjunctive Expressions-Prime Implicates
8. Irredundant Conjunctive Expressions and 2, 3 and 4 variable K-Map
9. The Quine-Mccluskey Method-To find Prime Implicants using decimal method upto 5 variables.
10. The Quine-Mccluskey Method-To find Prime Implicants using binary method upto 5 variables.

Unit – II

1. Introduction to Arithmetic Circuits & Data processing Circuits
2. Binary Adders & Subtracters, examples
3. Decimal Adders, examples
4. Code converter, examples
5. Full Adder and Full Subtractor, examples
6. Magnitude Comparators, examples
7. Multiplexers and Demultiplexers, examples
8. Decoders, examples
9. Parity generator and checkers, examples
10. Encoders, examples

Unit – III

1. Introduction to Memory and Programmable logic ,DAC Converters, VHDL Language
2. RAM, ROM, PROMs, PLA's, PAL
3. D/A Conversion and A/D Conversion: Variable
4. Resistor Networks
5. Binary Ladders
6. D/A Converters, D/A Accuracy and Resolution
7. VHDL – Introduction to VHDL
8. describing data flow, behavioural
9. describing structural and mixed design style
10. Simulating design for arithmetic and combinational circuits
11. Examples

Unit – IV

1. Introduction to Flip-Flops and Registers, Clock Waveform – Characteristics of ideal Clock Waveforms
2. Synchronous operation, propagational delay time
3. The Basic Bistable Element-Latches-Timing Considerations
4. K Master –Slave Flip-Flops
5. Pulse-Triggered Flip flops, Edge-Triggered Flip-Flops-Characteristic Equations
6. Conversions of Flip Flop
7. Types of Registers, Serial In – Serial Out, Serial In – Parallel out using JK or D Flip Flops
8. Parallel In – Serial Out, Parallel In – Parallel Out using JK or D Flip Flops
9. Applications of Shift Registers, Ring counter
10. Johnson counter
11. Sequence detector and sequence generator.

Unit – V

1. Introduction to Asynchronous and Synchronous Counter
2. Design of Synchronous up counter and down counter
3. Decade counter, Counter design as a synthesis problem
4. Design of Synchronous Sequential Circuits- Model Selection
5. State Transition Diagram, State Synthesis Table,
6. Design Equations and Circuit Diagram,
7. State Reduction Technique
8. Analysis of Asynchronous Sequential Circuit, Problems with Asynchronous Sequential Circuits
9. Design of Asynchronous Sequential Circuit
10. VHDL code for counters.

44.Explain Second –order Homogeneous Recurrence Relations

45.Solve the recurrence relation $a_n=3a_{n-1} – 2 a_{n-2}$ for n≥2 given that $a_1=5$ and $a_2=3$

46.Solve the recurrence relation $2a_{n+3} = a_{n+2} + 2a_{n+1} – a_n$ for n ≥ 0 with $a_0=0, a_1=1, a_2=2$

Lesson Plan

Unit – I

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 – II

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.

Unit – III

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, - minimal spanning trees-Fundamental cutsets and circuits.
7. The algorithms of Kruskal and Prim to find minimal spanning tree.
8. Finding minimal spanning tree by both algorithms.
9. Transport networks - Maxflow,Min-cut theorem –problems .

Unit – IV

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.

21. Prove that Every tree with n vertices has n-1 edges. 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.

22.How many internal vertices does complete 5 ary tree with 817 leaves have.How many leaves does a complete 6-ary tree of order 733 have.

23.Using merge sort method sort the list 7,3,8,4,5,10,6,2,9.

24. Explain the difference between DFS and BFS spanning trees of a graph with an example.

25. Write the steps involved in Prim's and Kruskal's algorithm.

26. Obtain an optimal prefix code for the message MISSION SUCCESSFUL . Indicate the code for the message 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

27. State Max flow and mincut theorem.

28. Find the number of nonnegative integer solutions of the equation $x_1+ x_2 + x_3+ x_4 = 18$ under the conditions $x_i \leq 7$ for i=1,2,3,4.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

29. State Max flow and mincut theorem.

30. Find the number of nonnegative integer solutions of the equation $x_1+ x_2 + x_3+ x_4 = 18$ under the conditions $x_i \leq 7$ for i=1,2,3,4.

31. Define Derangements and Find the number of derangements of 1,2,3,4

32. 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.

33 Explain Rook polynomial.

34.Find the rook polynomial for a 2x2 board by using the expansion formula.

35. Obtain the formula for d_n the number of derangements of n objects by using rook polymonials .

36.Find the sequences generated by the following functions $(3+x)^3$, $(1+3x)^{-1/3}$

37. Find a generating function for the following sequences
(i) 1,1,0,1,1,1,… (ii)0,2,6,12,20,30,42,….

38.Determine the coefficient of x^5 in the expansion of $(1-2x)^{-7}$

39. 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?

40.Explain Exponential generating function.

41. Solve the recurrence relation $a_{n+1} =4a_n$ for $n\geq0$ given that $a_0=3$

42. Find the recurrence relation and the intial condition for the sequence 0,2,6,12,20,30, 42……..

43.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 ?

Course Articulation Matrix (CAM)

Course Outcome (CO)	13	Program Outcome (ABET/NBA-(3a-k))												
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
Solve basic binary math operations using the logic gates.	**L1**	M	-	-	-	M	-	-	-	M	-	-	-	-
Demonstrate programming proficiency using the various logical elements to design practically motivated logical units.	**L3**	L	**H**	L	-	-	M	-	-	M	-	-	-	-
Design different units that are elements of typical computer's CPU.	**L4**	H	H	L	-	-	M	-	-	H	-	-	-	-
Apply knowledge of the logic design course to solve problems of designing of control units of different input/output devices.	**L3**	M	H	L	-	M	-	-	-	M	-	-	-	-
Explain the operation of basic building blocks of a flip-flop that is constructed of gates and latches.	**L3**	-	H	L	-	-	M	-	-	M	-	-	-	-
Analyse and design Asynchronous and Synchronous Sequential circuits.	**L1 & L4**	M	M	H	-	-	-	M	-	M	L	-	-	-

L- Low, M- Moderate, H-High

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**
Solve basic binary math operations using the logic gates.	**L1**	2	-	-	-	2	-	-	-	2	-	-	-	-
Demonstrate programming proficiency using the various logical elements to design practically motivated logical units.	**L3**	1	**3**	1	-	-	2	-	-	2	-	-	-	-
Design different units that are elements of typical computer's CPU.	**L4**	3	3	`	-	-	2	-	-	3	-	-	-	-
Apply knowledge of the logic design course to solve problems of designing of control units of different input/output devices.	**L3**	2	3	1	-	2	-	-	-	2	-	-	-	-
Explain the operation of basic building blocks of a flip-flop that is constructed of gates and latches.	**L3**	-	3	1	-	-	2	-	-	2	-	-	-	-
Analyse and design Asynchronous and Synchronous Sequential circuits.	**L1 & L4**	2	2	3	-	-	-	2	-	2	1	-	-	-

1 – Low, 2 – Moderate and 3 – High

Course Code : P13IS33	Semester : III	L - T - P : 4 - 0 - 0
Course Title : Data structures		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Computer Concepts and C Programming		

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 recursion .
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 and Implement different types of queues Using SLL.
7. Identify the different 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 – I

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 Hrs

Unit – II

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 Hrs

Unit – III

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 Hrs

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

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)
5. **Explain** Third and higher order Homogeous and non homogeneous recurrence relation.
6. **Obtain** the recurrence relation and initial condition for the given sequence.(L2)
7. **Solve** the second orderrecurrence relation using method of generating functions L3

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. Contruct a graph that is complete , regular, connected .
8. Differentialte 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.
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.

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

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)
6.**Explain** graph coloring problem, chromatic number, chromatic polynomial (L2)
7.**Find** the chromatic number and polynomial and partition of standard graphs.(L3)
8.**Determine** the chromatic number and polynomial for a given graph using
9. Decomposition and multiplication theorem(L3)

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

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)
6. **Explain** Dijkstras algorithm.(L1)
7. **Apply** Dijkstras algorithm to find the shortest path from single source to all other vertices.(L3)
8. **Explain and Apply** Prims and Kruskals algorithm to find the minimum spanning tree for the given graph. (L3)
9. Find the maximum flow and corresponding min-cut for the given transport network using max-flow Min –cut theorem.(L3)

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

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)
6. **Define** Generating functions.(L1)
7. **Find** the generating functions for the given sequence.(L3)
8. **Explain** different techniques for finding the generating function. (L3)
9. **Find** the number of partitions for a given positive integer using generating functions.(L3)
10. **Find** the exponential generating function for the given sequence. (L3)

Unit – IV

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 Hrs

Unit – V

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 Hrs

Text Book:

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

Reference Books:

1. "Fundamentals of Data Structures in C ", Horowitz, Sahani, Anderson-Freed , Second Edition, University Press .

Course Outcomes

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

1. Understand primitive and derived data structure.
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

1. Define data structure. – L1
2. Write an ADT specification for rational numbers and strings. – L2
3. List (classification) the different types of data structure. – L2
4. Explain with an example classification of data structure. – L2
5. Define postfix and prefix expression. – L1
6. Develop an algorithm to evaluate postfix and prefix expression. – L3
7. Develop an algorithm to convert infix to postfix and prefix to postfix. – L3
8. Write a recursive program to find Factorial of a Number, to generate nth Fibonacci Number and Tower of Hanoi. – L3
9. List the application of stack. – L2
10. Identify the differences between recursive and iterative programs. – L4

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

1. Define Static Memory Allocation and Dynamic Memory Allocation.– L1
2. List the differences between Static Memory Allocation and Dynamic Memory Allocation.– L 4
3. Identify the differences between array implementation and linked implementation.– L4
4. Define SLL, DLL, CSLL.– L1
5. Compare SLL and DLL.-L3
6. Write functions to perform basic operations on SLL.-L3
7. Write functions to perform basic operations on DLL.-L3
8. Write functions to perform basic operations on CSLL with header node.-L3
9. Write functions to perform basic operations on CSLL without header node. – L3
10. Develop a program to perform basic operations using above functions.– L3

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

1. Explain merging of two SLL, reversing a SLL, Searching an item in SLL with example.– L3
2. Develop an algorithm to merge two SLL, reversing a SLL, Searching an item in SLL.– L 3
3. Write a functions to merge the given two SLL reversing a SLL, Searching an item in SLL.– L 3
4. Given the polynomial, represent it using SLL.– L2
5. Explain the basic operations on linear queue with an example.– L 2
6. List the different methods to overcome the disadvantages of linear queue. – L2
7. Write a function for implementing queue using array and SLL.– L3
8. List the different types of queues.– L2
9. Explain the above types of queues with example.– L2
10. Write the function to implement basic operations on above queues.– L3

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

1. Define tree and the terms related to it.– L 1
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
6. Explain different tree traversal techniques.– L2
7. Write algorithms for different tree traversal techniques.– L3
8. Write binary tree for a given traversal sequences.– L3
9. Write the advantage of threaded binary tree.– L3
10. Write program to perform basic operations on threaded binary tree.– L3

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

1. Explain Radix sorting technique with example.-L2
2. Explain Merge sort technique.-L2
3. Write algorithms for different sorting techniques.– L2
4. List the applications of the above sorting techniques.– L3
5. Explain different searching techniques with example.– L2
6. Write algorithms for different searching techniques.– L 3
7. Which is the best searching technique given the large data set.-L3
8. List the applications of the above searching techniques.– L2
9. Compare different searching techniques.-L3
10. Compare different sorting techniques.-L3
11.

Unit – IV

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– calculation techniques, partitions of integers, The exponential generating function, The summation operator. **12hrs**

Unit – V

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

Third and higher –order Homogenous Recurrence relations, The non homogeneous recurrence relation, The method of generating functions for second order recurrence relations. **10hrs**

Text Books :

1. Discrete and Combinatorial Mathematics, RalphP. 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 parameters 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 .

Topic Learning Objectives

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

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)
5. **Explain** Hamilton cycle, path.(L2)
6. **Determine** a given graph has Hamilton path or cycle.(L3)
7. **Explain** Euler graphs with examples(L3)
8. **Identify** the different type of problems which lead to know the applications of graph theory(L2)
9. **Explain** Konigsberg bridge problem, travelling salesman problem, utility problem etc.

Course Code : P13IS45	Semester : IV	L - T - P : 4 - 0 - 0
Course Title : Graph theory and Combinatorics		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites :		

Course Learning Objectives (CLOs)

This course aims to

1. **Develop** the ability to identify different parameters of graphs.
2. **Understand** the simple applications of graph theory.
3. **Determine** if two graphs are Bipartite, to find the Hamilton path, cycle, finding the chromatic polynomial.
4. **Explain** Directed tree, rooted tree, binary rooted tree and the applications of rooted trees.
5. **Construct** optimal tree for the given prefix codes.
6. **Apply** Dijkstra's algorithm to find the shortest path from single source to all other vertices.
7. **Apply** Prim's and the Kruskal's algorithm to construct the minimal spanning trees.
8. **Apply** the principals of counting –rule of sum and product, permutations, combinations, Binomial theorem.
9. **Apply** the principles of inclusion and exclusion theorem, generalization principle for the given problem**.**
10. **Apply** the concept of generating functions. find the number of partitions of a positive integer for the given generating function.

Course Content

Unit – I

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 **10hrs**

Unit – II

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 hrs**

Unit – III

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 Hrs

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 n^{th} 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, and CSLL.
17. Write functions to perform basic operations on SLL, DLL, and 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 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.

Lesson Plan

Unit – I

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 – II

1. Linked Lists: 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 – III

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

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
Explain the basic structure and functioning of operating system.L2			**1**		**2**					**2**	**3**		
Solve problems related to process management and synchronization as well as able to apply learned methods to solve basic problems.L5	**3**	**2**	**3**		**2**							2	1
Understand the cause and effect related to deadlocks and is able to analyze them related to common circumstances in operating systems.L3	**2**	**1**	**3**		**2**					**1**	**3**	3	2
Explain the basics of memory management the use of virtual memory in modern operating system as well as the structure of the most common file system.L2	**2**		**1**		**2**						**1**	3	1
Describe how virtual memory works along with paging and page replacement algorithms.L4	**3**	**2**	**1**		**2**						**1**	2	1
Understand how the operating system abstractions can be used in the development of mass storage structure application programs.L3	2	1	3		1						1	1	1

L- Low, M- Moderate, H-High

. 9. Distinguish between file allocation methods. L4
10. solving problems related to file allocation L3
11. Explain the linked allocation method with merits and demerits L2

Unit – V

1. Discuss on secondary storage structure L2
2. Explain different disk scheduling algorithms L2
3. Problems solving on disk scheduling L3
4. Explain the need of swap space.
5. Illustrate access matrix l4
6. Discuss the goals of protection L2

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
Explain the basic structure and functioning of operating system.L2			**L**		**M**					**M**	**H**		
Solve problems related to process management and synchronization as well as able to apply learned methods to solve basic problems.L5	**H**	**M**	**H**		**M**							M	L
Understand the cause and effect related to deadlocks and is able to analyze them related to common circumstances in operating systems.L3	**M**	**L**	**H**		**M**					**L**	**H**	H	M
Explain the basics of memory management the use of virtual memory in modern operating system as well as the structure of the most common file system.L2	**M**		**L**		**M**						**L**	H	L
Describe how virtual memory works along with paging and page replacement algorithms.L4	**H**	**M**	**L**		**M**						**L**	M	L
Understand how the operating system abstractions can be used in the development of mass storage structure application programs.L3	M	L	H		L						L	L	L

L- Low, M- Moderate, H-High

Unit – IV

1. Trees-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;
9. Examples
10. Threaded Binary Trees-Threads
11. Inorder Traversal of a Threaded Binary Tree,
12. Inserting a Node into a Threaded Binary Tree.

Unit – V

1. Sorting Techniques-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.

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.	L2	-	-	H	-	M	-	-	M	H	-	M	-	H
Understand Abstract data types, Stacks and recursion.	L2	L	-	H	-	M	-	-	M	H	-	M	-	H
Develop and implement linked list.	L3	L	-	H	-	M	-	-	M	H	-	M	-	H
Develop programs to implement different queues.	L3	L	-	H	-	M	-	-	M	H	-	M	-	H
Understand and create trees.	L3	L	-	H	-	M	-	-	M	H	-	M	-	H
Understand and implement sorting and searching techniques.	L3	M	-	H	-	M	-	-	M	H	-	M	-	H
L- Low, M- Moderate, H-High														

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 primitive and derived data structure.	L2	-	-	3	-	2	-	-	2	3	-	2	-	3
Understand Abstract data types, Stacks and recursion.	L2	1	-	3	-	2	-	-	2	3	-	2	-	3
Develop and implement linked list.	L3	1	-	3	-	2	-	-	2	3	-	2	-	3
Develop programs to implement different queues.	L3	1	-	3	-	2	-	-	2	3	-	2	-	3
Understand and create trees.	L3	1	-	3	-	2	-	-	2	3	-	2	-	3
Understand and implement sorting and searching techniques.	L3	2	-	3	-	2	-	-	2	3	-	2	-	3
1 – Low, 2 – Moderate and 3 – High														

Lesson Plan

Unit – I

1. Explain the need of OS L2
2. Describe the structure of OS L2
3. List the operation of OS L1
4. Define distributed system L1
5. Discuss different types of system calls L2
6. Distinguish between peer-to-peer and client server systems L2
7. Explain the multithread model L2
8. Distinguish between process and thread L2
9. List the advantages of virtual machines L1

Unit – II

1. Demonstrate the concept of process L3
2. Explain process scheduling L2
3. List the criteria for process scheduling L1
4. Solve problems of process scheduling L3
5. Compare different process scheduling algorithms L4
6. Define process synchronization L1
7. Explain critical section problem l2
8. Explain the need of semaphores L2
9. Discuss different solutions for critical section problems L2.
10. Prepare monitor solution for dinning philosopher problem L5

Unit – III

1. Explain the necessary conditions for deadlock L2
2. List out the methods for handling deadlocks L1
3. Describe methods for recovering from deadlock L1
4. Discuss deadlock prevention methods L3
5. Solving problems occurring in deadlocks L3
6. Demonstrate the concept of paging L3
7. Explain the need of segmentation L2ement
8. Compare various memory management strategies L4
9. Explain the basic method of segmentation with respect to memory management.L2
10. Solving problems related to paging and segmentation L3
11. Differentiate between internal and external fragmentation L4

Unit – IV

1. Explain the need of virtual memory L2
2. Apply the concept of demand paging L3
3. Explain the different steps in handling page fault L2
4. Discuss the problems on page replacement algorithms L2
5. Solving problems related to page replacement L3
6. Explain the structure of file system L2
7. Describe the structure of directory L2
8. List the different file types with their functionsL1

21. What are the necessary and sufficient conditions for a dead lock to occur? Explain.
22. Describe any one method of recovery from deadlock.
23. Explain the basic method of Segmentation with respect to memory management.
24. Differentiate between internal & external fragmentation.
25. What is shared memory concept? Explain with example.
26. What is the purpose of paging the page tables.
27. Explain why sharing a reentrant module is easier when segmentation is used than when pure paging is used
28. Compare the circular waits scheme with the various deadlock avoidance scheme w.r.t the following issues A. Run time overhead B. System throughput
29. Why are segmentation and paging some times combined into one scheme.
30. Define i) Thrashing ii) Belady's anomaly iii)Effective access time in demand paging
31. Explain the different steps in handling page fault.
32. Consider following page reference string
1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.How many page faults would occur in case of LRU & optimal page replacement algorithm with 3 frames.
33. Explain paged memory allocation scheme and with a neat sketch explain paging hardware with TLB.
34. Explain the uses of virtual memory.
35. Explain the need of demand paging.
36. Analyze different page replacement algorithms.
37. Discuss the hardware support required to support demand paging.
38. Is it possible for a process to have two working sets one representing data and another representing code. Explain.
39. Explain different allocation methods.
40. Explain the structure of file system.
41. Explain various disc scheduling algorithms.
42. Define swapping.
43. Compare the performance of C scan and scan scheduling assuming a uniform distribution of request.
44. Explain why fairness is an important goal in time sharing system
45. What are the different goals to protect the system.
46. Explain the different principles of protection.
47. What are the three protection domain in a system.
48. Discuss the strength and weakness of implementing an access matrix using access lists that are associated with objects.
49. Explain the access matrix.

Course Code : P13IS34	**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 groups and its applications.
10. Apply coding theory concepts to code and encode a message.

Course Content

Unit – I

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 Hrs

Unit – II

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 Hrs

Unit – III

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 Hrs

Unit – IV

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 . Extremal elements , Lattices.

12 Hrs

Unit – V

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 Hrs

Text Book:

1. "Discrete and Combinatorial Mathematics", Ralph P. Grimaldi,B.V. Ramana , 5 th 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", Kolman Busby Ross , 5th edition , PHI.

Course Outcomes

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

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 and coding theory to solve the given problem.

9. Distinguish between file allocation methods. L4
10. solving problems related to file allocation L3
11. Explain the linked allocation method with merits and demerits L2

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

1. Discuss on secondary storage structure L2
2. Explain different disk scheduling algorithms L2
3. Problems solving on disk scheduling L3
4. Explain the need of swap space.
5. Illustrate access matrix l4
6. Discuss the goals of protection L2

Review Questions

1. What are the different components of computer system? Briefly explain them.
2. What is a system call ?Explain different types of system call.
3. Explain the features of distributed system. Hence distinguish between peer to peer and client server system
4. What is process and process control block? Explain different scheduling queues handled by OS?
5. What is IPC?
6. What are messages? How are they implemented?
7. Explain multithread model. Difference between user level thread and Kernel level thread.
8. Explain any five OS services.
9. What are the different models in Threads?
10. What are the main differences between main memory and secondary storage?
11. Differentiate between time sharing, multi-programming and batch processing system.
12. Explain the following i) Short term scheduler Long-term scheduler iii) Virtual machine.
13. For the following example calculate average turn around time, average waiting time for following algorithms
    i) FCFS ii) preemptive SJF i) Round Robin (1 time unit)
14. Differentiate between short term and long term scheduler.
15. Explain different scheduling criteria used for comparing CPU scheduling algorithms
16. What is critical section problem? Write and explain two process solutions
17. What is Dinning philosophers problem? Write and explain monitor solution for Dinning philosophers problem.
18. With the neat diagram explain process state transition.
19. Explain why spin locks are not appropriate for single processor system
20. Explain the differences in the degree to which the following scheduling algorithms discriminate in favour of short processes A. FCFS B.RR

Topic Learning Objectives

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

1. Explain the need of OS L2
2. Describe the structure of OS L2
3. List the operation of OS L1
4. Define distributed system L1
5. Discuss different types of system calls L2
6. Distinguish between peer-to-peer and client server systems L2
7. Explain the multithread model L2
8. Distinguish between process and thread L2
9. List the advantages of virtual machines L1

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

1. Demonstrate the concept of process L3
2. Explain process scheduling L2
3. List the criteria for process scheduling L1
4. Solve problems of process scheduling L3
5. Compare different process scheduling algorithms L4
6. Define process synchronization L1
7. Explain critical section problem l2
8. Explain the need of semaphores L2
9. Discuss different solutions for critical section problems L2.
10. Prepare monitor solution for dinning philosopher problem L5

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

1. Explain the necessary conditions for deadlock L2
2. List out the methods for handling deadlocks L1
3. Describe methods for recovering from deadlock L1
4. Discuss deadlock prevention methods L3
5. Solving problems occurring in deadlocks L3
6. Demonstrate the concept of paging L3
7. Explain the need of segmentation L2ement
8. Compare various memory management strategies L4
9. Explain the basic method of segmentation with respect to memory management.L2
10. Solving problems related to paging and segmentation L3
11. Differentiate between internal and external fragmentation L4

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

1. Explain the need of virtual memory L2
2. Apply the concept of demand paging L3
3. Explain the different steps in handling page fault L2
4. Discuss the problems on page replacement algorithms L2
5. Solving problems related to page replacement L3
6. Explain the structure of file system L2
7. Describe the structure of directory L2
8. List the different file types with their functionsL1

Topic Learning Objectives

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

1. Solve the problems of counting theory applying the rules of sum and product.-L2
2. Differentiate between Permutations and Combinations, Combinations with repetition 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
6. Using laws of set theory , Membership table method and Venn diagram method Prove that the given two representations of sets are equal or not. -L2
7. Using addition principle for three sets and more Solve the problems of counting theory.-L2
8. Identify the laws of set theory and their uses.-L2
9. Apply Laws of set theory to represent a given set in another form.-L3
10. Define probability and find Probability of given event (using addition principle .-L2

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

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.-L
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
6. Define Quantifiers, Express the statement in the symbolic form, negate the statement.-L1
7. Find the truth value of the given statement and write its negated form .-L3
8. Check the validity of the argument (with a quantified statement) expressing it in the symbolic form.-L3
9. Identify and find the type of proofs of theorems (Direct and indirect methods- contrapositive and contradiction methods.-L3
10. Find the proof of the given statement in direct or indirect method.-L2

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

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
6. Define Relations ,functions- one-one and onto functions.-L1
7. Apply Stirling's Number of second kind to solve problems.-L3
8. Find the number of one –one functions, onto functions, bijective functions.-L1
9. Solve the problems using pigeon hole principle.-L2
10. Define special functions-characteristic, permutation, hashing functions properties. Domain , Co domain and Range of each type of function.-L1

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

1. Define the properties of relations, represent the relation in the matrix form Digraph form, and Identify the relation given in any form.-L1
2. Prove that the given relation in an equivalence relation or partially ordered relation.-L3
3. Define Hasse diagram.-L1
4. Construct the Hasse diagram for a given partially ordered relation.-L3
5. Given the Hasse diagram find the number of elements present in the set and the relation. -L3
6. Define the properties of an equivalence relation.-L1
7. State and Prove the theorem listing the properties.-L3
8. Find the partition induced by an equivalence relation with properties for a given relation.-L2
9. Find the external elements of a given relation- maximal, minimal, least ,greatest element ,GLB, LUB of a subset of the given set.-L3
10. Define a lattice, properties, recognize it in any form.-L1

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

1. Define a Group –Examples ,properties Recognize the properties of a Group, subgroup, cyclic group .-L1
2. State Lagrange's theorem.-L1
3. Prove Lagrange's theorem.-L2
4. Define homomorphism, Isomorphism and cyclic groups ,cosets.-L2
5. Elementary type of groups-relations between them.-L1
6. Define elements of coding theory and Hamminng metric.-L1
7. Define Pariy check matrix, Generator matrices, coding and decoding with coset leaders.-L1
8. Define Hamming code.
9. Find Encode and Decode the given messages. Given the parity check matrix for a Hamming code.-L2
10. Construct a decoding table for group code given by the generator matrix.-L2

Unit – II

PROCESS SCHEDULING AND PROCESS SYNCHRONIZATION: Basic concepts of process scheduling; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling. Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. **10 Hrs**

Unit – III

DEADLOCKS: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **MEMORY MANAGEMENT:** Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

10hrs

Unit – IV

VIRTUAL MEMORY MANAGEMENT, FILE SYSTEM AND IMPLEMENTATION OF FILE SYSTEM: Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management. **11Hrs**

Unit – V

SECONDARY STORAGE STRUCTURES, PROTECTION: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability-Based systems, case study. **10 Hrs**

Text Book:

1. Operating System Principles – Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 7th edition, Wiley-India, 2006.

Reference Books:

1. **Operating Systems: A Concept Based Approach** – D.M Dhamdhere, 2nd Edition, Tata McGraw- Hill, 2002.
2. **Operating Systems** – P.C.P. Bhatt, 2nd Edition, PHI, 2006.
3. **Operating Systems** – Harvey M Deital, 3rd Edition, AddisonWesley, 1990

Course Outcomes

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

1. Explain the basic structure and functioning of operating system.
2. Solve problems related to process management and synchronization as well as able to apply learned methods to solve basic problems.
3. Understand the cause and effect related to deadlocks and is able to analyze them related to common circumstances in operating systems.
4. Explain the basics of memory management the use of virtual memory in modern operating system as well as the structure of the most common file system.
5. Describe how virtual memory works along with paging and page replacement algorithms.
6. Understand how the operating system abstractions can be used in the development of mass storage structure application programs.

Course Code : P13IS44	Semester : IV	L - T - P : 4 - 0 - 0
Course Title : **Operating System**		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50	
Prerequisites : Computer Organization and OOPs concpt		

Course Learning Objectives (CLOs)

This course aims to

1.To enable the Knowledge of operation, implementation and performance of modern operating systems, and the relative merits and suitability of each for complex user applications

2.To understand the basic operations and services provided by operating system

3.To demonstrate the knowledge of process and process synchronisation

4.To discuss and explain CPU scheduling and its relevance to operating systems

5.To Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software

6. To explain what deadlock is in relation to operating systems and also describe deadlock detection and recovery

7.T o describe how virtual memory works along with paging and page replacement algorithms.

8. To explain a file system and various file allocation methods.

9.To Understand how the operating system abstractions can be used in the development of Mass storage structures application programs, or to build higher level abstractions

10.To compare, contrast, and evaluate the key trade-offs between multiple approaches to operating system design, and identify appropriate design choices when solving real-world problems.

11.To Understand basic resource management techniques (scheduling or time management,

Course Content

Unit – I

introduction to operating systems, system structures and process concepts: Need of operating systems; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special purpose systems; Computing environments. Operating System Services; User- Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines. Process concepts, Process scheduling, operations on processes, Inter -process communication. Multi-Threaded Programming: Overview; Multithreading models. **11 Hrs**

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. Find the number of ways of distributing seven apples and six oranges among 4 students such that each get atleast 1 apple.
5. Find the coefficient of $x^2 y^3 z^2$ in the expantion $(x+y+z)^7$
6. 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
7. 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?
8. If two integers are selected at random and without replacement from {1,2, …,99,100} what is the probability that their sum is even.
9. If a fair coin is tossed tour times what is the probability that two heads and two tails occur.
10. 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.
11. Define logical equivalence and using laws verify $(\neg p \vee q) \wedge (p \wedge (p \wedge q)) \equiv (p \wedge q)$
12. Express symbolically and check the validity. It is not sunny this afternoon and is colder than 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.
13. 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
14. Define Rule of universal specification and generalization.
15. Expressing symbolically check the validity" No junior or senior has enrolled in sports. Raju has enrolled in sports. Therefore, Raju is not a senior."
16. Prove or disprove directly "The sum of any five consecutive integers is always divisible by 5".
17. State mathematical Induction principle and Prove that $1.3+2.4+\ldots\ldots+n(n+2)= n(n+1)(2n+7)/ 6$ for all integers $n\geq 1$
18. (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 $n \geq 1$
19. Define one-one functions, onto functions with example for each.
20. Find the number of one-one and onto functions from a set of m elements to a set of n elements.
21. State Pegion hole principle and extended Pegion hole principle.
22. Prove that any subset of size 6 from the set S={1,2,3,…,9}must contain two elements whose sum is 10.
23. 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 $f^{-1}o\, g^{-1}$
24. Write the formula to find p(m ,n), S(m ,n)and p(m) . what does each number represent Counting theory.

25. Define domain, codomain and range of a given function, Justify each with reason.
26. Explain Characteristic function, permutation function and hashing function and their uses.
27. Let R be a relation defined as “exactly divides” on A={1, 3, 6, 9,11, 35, 385}
    i) Is R a Poset ,verify
    ii) Draw the Hasse diagram of the poset .
28. Draw Hasse diagram of all positive divisors of 36
29. Define least, greatest, maximal, minimal element in a poset.
30. 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 .
31. 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.
32. Prove that [M(R)]2 = M (R2) for a given relation R on A.
33. Prove that [x] = [y]or [x]∩[y]=Ø for any two elements of a poset (A,R).
34. Define Lower bound and upper bound, GLB, LUB of a subset of a poset.
35. Find GLB, LUB of the subset of a poset whose hasse diagram is given.
36. How to convert a partially ordered set into a totally ordered set?
37. Define a Lattice with an example.
38. Represent a Lattice in a digraph form.
39. Prove that (A, “subset of”) is a poset.
40. Define a Group –Examples, properties List the properties of a Group, subgroup, and cyclic group.
41. State and prove Lagrange’s theorem.
42. Define homomorphism, Isomorphism and cyclic groups , cosets.
43. Define, Homomorphism, Isomorphism between two groups with an example.
44. Write short notes on Encoding and Decoding of a message.
45. Define Generator Matrix, Parity-check Matrix.
46. 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.
47. For an encoding function E:$Z_2^{\ 4} \rightarrow Z_2^{\ 6}$ is defined by the generator matrix G

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

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

(ii) Determine the associated parity check matrix.

Course Articulation Matrix (CAM)

Course Outcome (CO)		Program Outcome												
		a	b	c	d	e	f	g	h	i	j	k	l	m
Design finite automata.		2	3	2	-	2	-	-	-	2	-	-	2	2
Explain equivalence and minimization of finite automata.		-	1	1	-	-	-	-	-	2	-	-	2	2
Design regular expression for regular languages, convert between finite automata and regular expressions for regular languages.		2	1	1	-	-	-	-	-	2	-	1	2	2
Design grammars for various languages		-	2	2	-	-	-	-	2	-	-	-	2	2
Design push-down automata from grammars.		-	2	2		-	-	-	-	2	-	-	2	2
Design Turing machines for simple languages and functions.		-	2	2	-	-	-	-	-	2	-	-	2	2
Design problem reductions to determine the un-decidability of languages		-	2	1	-	-	-	-	-	2	-	-	2	2

L- Low, M- Moderate, H-High

Course Articulation Matrix (CAM)														
Course Outcome (CO)		**Program Outcome**												
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
Design finite automata.		**M**	**H**	**M**	**-**	**M**	**-**	**-**	**-**	**M**	**-**	**-**	**M**	**M**
Explain equivalence and minimization of finite automata.		**-**	**L**	**L**	**-**	**-**	**-**	**-**	**-**	**M**	**-**	**-**	**M**	**M**
Design regular expression for regular languages, convert between finite automata and regular expressions for regular languages.		**H**	**L**	**L**	**-**	**-**	**-**	**-**	**-**	**M**	**-**	**L**	**M**	**M**
Design grammars for various languages		**-**	**M**	**M**	**-**	**-**	**-**	**-**	**M**	**-**	**-**	**-**	**M**	**M**
Design push-down automata from grammars.		**-**	**M**	**M**	**-**	**-**	**-**	**-**	**-**	**M**	**-**	**-**	**M**	**M**
Design Turing machines for simple languages and functions.		**-**	**M**	**M**	**-**	**-**	**-**	**-**	**-**	**M**	**-**	**-**	**M**	**M**
Design problem reductions to determine the un-decidability of languages		**-**	**M**	**L**	**-**	**-**	**-**	**-**	**-**	**M**	**-**	**-**	**M**	**M**
L- Low, M- Moderate, H-High														

Lesson Plan

Unit – I

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 – II

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

Unit – III

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 – IV

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 R2 ,R3 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, External elements of the poset.
9. Toset and its properties, Define Lattice.
10. LUB, GLB of a subset of a POSET
11. Properties of a lattice
12. Problems

Unit – V

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

7. Regular languages.
8. Proving languages not to be regular languages.
9. Closure properties of regular languages.
10. Decision properties of regular languages;
11.Review.

Unit – III

1.Definition of grammar and its component.
2.Definition of CFG and type of grammar
3.Context -free grammars.
4. Applications of CFG.
5. Parse trees, leftmost derivation and right most derivation.
6. Ambiguity in grammars.
7. Design CFG for some languages.
8. Prove that Language is not CFL using pumping lemma.
9. Closure properties of CFL.
10. Continuation of Closure properties of CFL.
11. Normal forms of CFG.
12. Review.

Unit IV

1. Definition of the Pushdown automata.
2. The languages of a PDA.
3. Design of PDA for some languages
4. Continuation of Design of PDA.
5. Equivalence of PDA's and CFG's
6. Deterministic Pushdown Automata.
7. Difference b/w Deterministic Pushdown Automata non- Deterministic Pushdown Automata.
8. Review.

Unit V

1. Problems that Computers cannot solve
2. The turning machine.
3. Programming techniques for Turning Machines.
4. Extensions to the basic Turning Machines.
5. Turing Machine and Computers
6. A Language that is not recursively enumerable
7. An un-decidable problem that is RE.
8. Post's Correspondence problem.
9. Other un-decidable problems.
10. Review.

7. Conversion of Finite *automata to Regular expressions using state elimination method or kleens theorem.*
8. Application of RE.
9. What do you mean by regular languages?
10. What is the closure property of regular languages
11. Problems solving regarding Minimization of DFA.
12. Context free languages are described by type _____ grammar.
13. Grammar S □ aSb | SS | ε is _____ grammar.
14. _____ languages are the subset of Context free languages.
15. Syntax of any high level language is defined with _____.
16. How the transistion/move of a PDA defined.
17. What are the demerits of regular languages when compared to context free languages.
18. What are the demerits of DFA when compared to PDA.
19. Obtain a PDA to accept the language L={ $a^n b^n$ | n>=0}
20. For every regular language we can construct _____ to accept the languages.
21. In PDA, stack provides _____ memory.
22. What do you mean by Turing machine.
23. Languages accepted by Turing machine.
24. What do you mean by undeciadable problem.
25. Problems which have yes or no answer are called decision problem.
26. _____ are those problem that can be solved by a Turing machine in polynomial time.

Lesson Plan

Unit – I

1. Introduction to Finite Automata
2. The central concepts of Automata theory.
3. Design of Deterministic finite automata.
4. Design of Non-Deterministic finite automata.
5. Conversion of DFA to NFA using subset construction.
6. Conversion of DFA to NFA using Lazy evaluation method.
7. Differences b/w NFA and DFA
8. Design of epsilon NFA
9. Conversion of epsilon NFA to DFA
10. Equivalence and minimization of automata using table filling algorithm.
11. Applications of Finite automata.
12. Review.

Unit – II

1. An application of finite automata.
2. Finite automata with Epsilon transitions.
3. Regular expressions.
4. Finite Automata and Regular Expressions.
5. Applications of Regular Expressions.
6. Conversion Finite automata to Regular expressions using state elimination method or kleens theorem.

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 principles of counting and set theory	L2	H	L	-	L	L	-	-	L	-	-	-	M	-
Identify the quantifiers and their uses and learn the fundamentals of logic theory	L3	H	L	-	-	L	M	H	-	-	-	-	L	-
Apply the Mathematical induction principle and pegion hole principle to solve the real time problems	L4	H	L	-	-	L	-	-	L	-	-	-	L	-
Solve the problems Using the concepts of relations and functions and Identify the different ways of representing relations	L3	H	L	-	-	-	-	-	-	-	-	-	L	-
Apply the concepts of group theory and coding theory to solve the given problem	L5	H	L	-	-	L	-	-	-	-	-	L	L	-

L- Low, M- Moderate, H-High

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 principles of counting and set theory	L2	3	1	-	1	1	-	-	1	-	-	-	2	-
Identify the quantifiers and their uses and learn the fundamentals of logic theory	L3	3	1	-	-	1	2	3	-	-	-	-	1	-
Apply the Mathematical induction principle and pegion hole principle to solve the real time problems	L4	3	1	-	-	1	-	-	1	-	-	-	1	-
Solve the problems Using the concepts of relations and functions and Identify the different ways of representing relations	L3	3	1	-	-	-	-	-	-	-	-	-	1	-
Apply the concepts of group theory and coding theory to solve the given problem	L5	3	1	-	-	1	-	-	-	-	-	1	1	-

1 – Low, 2 – Moderate and 3 – High

Course Code : P13IS35	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 : Digital Design and Computer Concepts and C Programming		

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 and Interrupts,
4. Understand the concept DMA and Exceptions.
5. Explain different types of memories with their functionalities.
6. Understand the concept of virtual & cache memory
7. Understand the design and working of fast adders.
8. Understand different algorithms for performing arithmetic operations.
9. Explain the concept of bus organization, pipelining and multiprocesssors.
10. Understand the different types in generation of control signals.

Course Content

Unit – I

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 Hrs

Unit – II

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 Hrs

Unit – III

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

10 Hrs

9. convert automata to regular expression L2
10. converting Finite Automata to Regular Expression using Kleen's Theorem L2

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

1. Define Grammar and its component. L1
2. Design of CFG for some languages. L2
3. Applications of CFG L3
4. Prove the closure properties of CFG L2
5. Learn about CNF and conversion of grammar to different normal forms L1
6. Learn about GNF and conversion of grammar to different normal forms L1
7. Construction of parse tree L2
8. Find out Ambiguity in grammars. And removal of ambiguity in the grammar. L2
9. Prove that the grammar is not CFL. L3
10. Define parse Tree L1

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

1. Know languages accepted by push down automata. L1
2. Definition of PDA. L1
3. Design an PDA for some Language L3
4. Theorems that show equivalence of PDA and Context free languages. L2
5. Conversion of CFG to PDA L3
6. Differentiate between Deterministic PDA and NON-Determinstic PDA L3
7. Conversion of PDA to CFG L3
8. Design of Deterministic PDA. L3

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

1. Definition of Turing machine. L1
2. Discussion about multitape Turing machine L2
3. Design of Turing machine to accept languages. L3
4. Non deterministic Turing machine. L1
5. Understand meaning of recursively enumerable languages. L1
6. Understand Post's Correspondence problem. L1
7. Understand the undecidability problems L1
8. Understanding the meaning of universal languages L1

Review Questions

1. Design a DFA to accept strings of a's and b's having not more than three a's.
2. Design a DFA to accept strings of a's and b's having even numbers of a's and even numbers of b's.
3. Explain the applications of Finite automata.
4. Problems on Conversion of NFA to DFA
5. Meaning of some of the meta character like *,. ,+ in unix and equivalent RE.
6. Write down the RE for some languages.

Unit – V

Introduction to Turing Machine, Un-decidability Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines; Extensions to the basic Turning Machines; Turing Machine and Computers. Un-decidable problem that is RE; Post's Correspondence problem.

10hrs

Text Books:

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

References:

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.
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.Design finite automata. L4
2.Explain equivalence and minimization of finite automata. L2
3.Design regular expression for regular languages, convert between finite automata and regular expressions for regular languages. L3
4.Design grammars for various languages L3
5.Design push-down automata from grammars. L2
6.Design Turing machines for simple languages and functions. L3
7.Design problem reductions to determine the un-decidability of languages. L3

Topic Learning Objectives

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

1.Describe some basic concepts of set theory and relations. Prove theorem based on Induction, contradiction. L2
2.Design of automata for pattern recognition, modulo-K counter problem and divisible by k problems. L3
3.Applications of finite automata. L3
4.Explain the difference between NFA and DFA.L3
5.Conversion of NFA to DFA. L3
6.Conversion of NFA to DFA using subset construction or lazzy evaluation method L3
7.Explain the difference between NFA and ε-NFA. L2
8.Reduce the numbers of states using table filling algorithm L3
9.Conversion of ε-NFA to DFA. L3
10.Design an NFA for some languages. L3

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

1.Define Regular expressions. L1
2.Write down Regular expressions for some languages. L2
3.Convert Regular expressions to automata and vice-versa. L2
4. Converting Finite automata to Regular expressions using state elimination method. L2
5.Applications of Regular expressions in UNIX. L3
6.Know about regular languages.L1
7.Proving languages are not regular by applying pumping lemma theorem. L3
8.Know about Closure properties of regular languages. L1

Unit – IV

Arithmetic: Addition & subtraction of signed numbers, Design of fast adders; Multiplication of positive numbers, Signed-operand multiplication, Fast multiplication, Integer division, Floating point numbers and operations.

10 Hrs

Unit – V

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 Hrs

Text Book:

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

Reference Books:

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

Course Outcomes

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

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 execution of instruction with different bus architecture.
6. Understand the operation of pipelining and multiprocessor.

Topic Learning Objectives

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

1. Understand the basic structure of a computer.-L2
2. Write machine instructions and understand their execution including branching, and subroutine call and return operation.-L3
3. Understand the performance issues in computer systems.-L2
4. Explain Number representation and addition/subtraction in the 2's complement system.-L2
5. Identify different Addressing methods for accessing register and memory operands.-L1
6. Understand how Program control input / output operations are performed. –L2
7. Explain different Operations on stack, queue. –L2
8. List out and perform different Shift operation types. –L1
9. Understand the concept of Encoding-1, 2, 3 word instructions.-L2
10. Understand the concept of Subroutines. –L2

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

1. Understand how Program controlled I/O is performed using polling.-L2
2. Understand the idea of interrupts and the hardware and the software needed to support them.-L2
3. Explain direct memory access I/O mechanism for high speed devices.-L2
4. Differentiate between data transfer over synchronous and Asynchronous buses.-L4
5. Understand the design of I/O interface circuits.-L2
6. Identify Commercial bus standards in particular PCI, SCSI, USB buses.-L2
7. Explain the various ways to perform I/O operations-L2
8. Differentiate between centralized & distributed bus arbitration.-L4
9. Explain the architecture of USB.-L2
10. Differentiate between subroutine and ISR. –L4

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

1. Understand the concept of Basic memory circuits.-L2
2. Explain Organization of the main memory.-L5
3. Understand Cache memory concept, which shortens the effective memory access time.-L2
4. Understand Virtual memory mechanism, which increases the apparent size of the main memory. –L2
5. List the various types of ROM-L1
6. Explain the different techniques available for mapping in cache memory – L2
7. Understand how the performance increases with respect to cache—L2
8. write the organization of a memory module using static RAM chip of different sizes—L3
9. Differentiate b/w asynchronous & synchronous DRAMS –L4
10. Explain the internal organization of memory chip –L2

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

1. Design High speed adders implemented in a hierarchical structure using carry look ahead logic to generate carry signals in parallel.-L5
2. Apply the Booth algorithm to determine how multiplicand summands are selected by the multiplier bit patterns in performing multiplication of signed numbers.-L3
3. Design Circuits that perform division operations.-L5
4. Explain the representation of floating point numbers in IEEE standard format and how to perform basic arithmetic operations on them. –L2
5. Explain the floating point operations—l2
6. Design full adder using half adders & external logic gates –L5
7. Understand how integer division is performed using restoring algorithm—L2
8. Design Circuits that perform division operations.-L3
9. Explain the representation of floating point numbers in IEEE standard format and how to perform basic arithmetic operations on them.-L2
10. Understand how integer division is performed using non restoring algorithm—L2

Course Code : P13IS43	Semester : IV	L - T - P : 4 - 0 - 0
Course Title : Finite Automata and Formal language		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50	
Prerequisites :		

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.
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 un-decidability of languages.

Course Content

Unit – I

Introduction to Finite Automata, Regular Expression

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. **10hrs**

Unit – II

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 hrs**

Unit – III

Context-Free Grammars And properties of Context-Free Languages: Context –free grammars; Parse trees; Applications; Ambiguity in grammars and Language**s,** Definitions of Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFLs. **12 hrs**

Unit – IV

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

10 hrs

Course Articulation Matrix (CAM)														
Course Outcome (CO)		**Program Outcome**												
		a	b	c	d	e	f	g	h	i	j	k	L	m
Analyze the space and time complexities for the given problem.		3	1	1										
Solve problems on searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer.		3	2	2		2		1		2		3		2
Solve graph based problems using the different algorithm techniques.		3	2	2		2			1	2		3		2
Apply solutions to overcome the limitations of algorithms		3				2					3			2
Identify and Apply algorithm Techniques to solve realistic problems.		3	2	2						2		3		2
Design and **implement** algorithms for the given problem.		3	2	3		2		3		2		3		
Low-1, Moderate-2, High-3														

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

1. Analyze how a processor executes instructions.-L4
2. Understand the internal functional units of a processor and how they are inter connected.-L2
3. Design Hardware for generating internal control signals.-L5
4. Explain the microprogramming approach.-L2
5. Understand Micro program organization.-L2
6. Design Hardware for generating internal control signals.-L3
7. Explain the microprogramming approach.-L2
8. Understand Micro program organization.-L2
9. Understand the basic concepts of pipelining –L2
10. Explain the structure of multiprocessor-L2

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.
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. Discuss the stack structure and its operations.
11. In modern computers why interrupts are required? Support your claim with suitable example.
12. In the interrupt mechanism, how simultaneous arrivals of interrupts from various devices are handled.
13. Define bus arbitration. List and explain various approaches to Bus arbitration.
14. Explain, with the help of a diagram the working of daisy chain with multiple priority levels and multiple devices in each level.
15. Define exceptions. Explain 2 kinds of exceptions.
16. Explain the following: (i) Interrupt concepts (ii) Interrupt hardware.
17. Define cycle stealing, Burst mode.
18. With a neat sketch explain the individual input and output interface circuits. Also list their salient features.
19. In the computer system why PCI bus is used.
20. Explain with a block diagram a general 8 bit parallel interface.
21. Define and explain the following: (i) Memory access time (ii) Memory Cycle time (iii) Random access memory (iv) Static memory.
22. Differentiate between the static RAM and Dynamic RAM giving 4 key differences. State the primary usage of SRAM and DRAM in contemporary computer systems.
23. Explain a simple method of translating virtual address of a program into the physical, with the help of a diagram.
24. 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.
25. With the block diagram explain the operation of a 16 bit megabit DRAM configured as 2MX8.
26. Explain different mapping functions used in the cache memory.
27. What is memory interleaving? Explain.
28. Write the organization of 1K X 1 memory chip
29. Write briefly about ROM
30. Explain the operation in cache memory

Course Articulation Matrix (CAM)

Course Outcome (CO)	Program Outcome												
	a	b	c	d	e	f	g	h	i	j	k	l	m
Analyze the space and time complexities for the given problem.	H	L	L										
Solve problems on searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer.	H	M	M		M		L		M		H		M
Solve graph based problems using the different algorithm techniques.	H	M	M		M			L	M		H		M
Apply solutions to overcome the limitations of algorithms	H				M					H			M
Identify and Apply algorithm Techniques to solve realistic problems.	H	M	M						M		H		M
Design and **implement** algorithms for the given problem.	H	M	H		M		H		M		H		

L- Low, M- Moderate, H-High

6. Space and time tradeoffs,
7. sorting by counting,
8. Sorting by distribution,
9. Input enhancement in string matching,
10. Hashing.

Unit – IV

1. Dynamic programming – introduction,
2. Warshall's algorithm
3. Floyd's algorithms,
4. Binomial coefficient,
5. the knapsack problem and memory functions,
6. Greedy technique – Introduction
7. Prim's algorithm,
8. Kruskal's algorithm,
9. Dijkstra's algorithm,
10. Huffman's trees

Unit – V

1. Lower-bound arguments,
2. Decision trees,
3. P, NP, and NP-complete problems,
4. Challenges of numerical algorithms,
5. coping with the limitations of algorithm power – Backtracking, Branch and bound,
6. Approximation algorithms for NP-hard problems,
7. Algorithms for solving nonlinear equations.

31. Explain the concept of carry save addition for the multiplication operation, MXQ=P for 4-bit operands, with diagram and suitable example.
32. In a carry look ahead adder explain the generate Gi and Propagate Pi function for stages with the help of Boolean expression for Gi and Pi.
33. 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.
34. Perform the division of 8 by 3 using non-restoring division algorithm.
35. Explain how a 16-bit carry look ahead adder can be built from 4 bit adder.
36. Show the multiplication of (+13) and (-6) using multiplier bit pair recoding technique.
37. Differentiate between restoring and non-restoring division algorithms.
38. Explain the IEEE standard for floating point number representation.
39. Show how to implement a full adder using half adders & external logic gates
40. Represent the following decimal numbers using IEEE standard floating point notation for +1.725
41. Explain the process of fetching the word from the memory using timing diagram of memory read operation. Also give an example for the same.
42. Write the control sequence of execution of the instruction ADD (R3),R1 using single bus organization.
43. Write and explain the control sequence for the execution of an unconditional branch instruction.
44. Explain with block diagram the basic organization of a microprogrammed control unit.
45. What are the modifications required in the basic organization of a microprogrammed control unit to support conditional branching in the microprogram.
46. Explain with diagram how control signals are generated using single bus organization.
47. Explain the multibus organization.
48. Draw the block diagram of a complete processor and identify the units.
49. Explain the basic concepts of pipelining
50. Explain the structure of multiprocessor

Lesson Plan

Unit – I

1. Computer types, Functional units
2. Basic operational concepts, Bus structures
3. Performance
4. Numbers, arithmetic operations & characters,
5. Memory location & addresses, Memory operations
6. Instructions & instruction sequencing
7. Instructions & instruction sequencing
8. Addressing modes
9. Assembly language, Basic input/output operations
10. Stacks & queues, Subroutines
11. Additional instructions, Encoding of machine instructions

Unit – II

1. Accessing I/O devices
2. Interrupts-Interrupt hardware, Enabling and Disabling Interrupts

3. Handling Multiple Devices
4. Controlling Device Requests
5. Exceptions
6. Direct memory access
7. Buses
8. Interface circuits
9. Standard I/O Interfaces
10. Standard I/O Interfaces

Unit – III

1. Basic concepts
2. Semiconductor RAM memories
3. Semiconductor RAM memories
4. Read-Only memories
5. Cache memories-Mapping Functions
6. Cache memories-Mapping Functions
7. Replacement Algorithms
8. Replacement Algorithms
9. Performance considerations
10. Virtual memory

Unit – IV

1. Addition & subtraction of signed numbers
2. Design of fast adders
3. Design of fast adders
4. Multiplication of positive numbers
5. Signed-operand multiplication
6. Fast multiplication
7. Integer division
8. Integer division
9. Floating point numbers and operations
10. Floating point numbers and operations

Unit – V

1. Basic processing unit: Some fundamental concepts
2. Some fundamental concepts
3. Execution of a complete instruction
4. Multiple bus organization
5. Hardwired control
6. Micro programmed control
7. Micro programmed control
8. Basic concepts of pipelining
9. Basic concepts of pipelining
10. The structure of general purpose multiprocessors
11. Memory organization in multiprocessors

37. Solve the Problems using Kruskal's Algorithm? Derive the Complexity of the Algorithms.
38. Solve the Problems using Prim's Algorithm? Derive the Complexity of the Algorithms.
39. Explain Coding Technique? Identify the requirements of Coding Problems.
40. Give an Example for Coding Problem? Solve the Problems using Huffman Coding Algorithm? Derive the Complexity of the Algorithm.
41. Define Backtracking? State the principle of Backtracking?
42. What are the applications of backtracking?
43. State if Backtracking always produces optimal solution?
44. Explain n-queens problem? Derive the time complexity for n-queens problem.
45. What is branch and bound?
46. Compare backtracking & branch and bound technique?

Lesson Plan

Unit – I

1. What is an algorithm? Fundamentals of algorithmic problem solving.
2. important problem types,
3. Fundamental data structures,
4. Fundamentals of the analysis of algorithm efficiency
5. Analysis framework,
6. asymptotic notations
7. basic efficiency classes,
8. Mathematical analysis of recursive algorithms.
7. Mathematical analysis of Non-recursive algorithms.
8. Example of designing algorithms

Unit – II

1. Brute force – introduction, selection sort
2. bubble sort, Sequential search
3. String matching. Divide and conquer – introduction
4. Merge sort, Quick sort, Binary search, Binary tree traversals and related properties,
5. Multiplication of large integers
6. Matrix multiplication
7. Decrease and conquer – introduction,
8. insertion sort,
9. depth-first search
10. breadth-first search
11. Topological sorting

Unit – III

1. Transform and conquer – introduction
2. Presorting,
3. Balanced search trees,
4. Heap,
5. Heap sort,

11.Applying the concept of Brute Force to String matching? Analysis of the algorithm
12. Define the terms associated with Divide and Conquer strategy
13. Applying the concept of Divide and Conquer to Merge Sort? Analysis of the algorithm
14. Applying the concept of Divide and Conquer to Quick Sort? Analysis of the algorithm
15. Understand the working of Binary Search algorithm? Analysis of Binary Search Algorithm.
16. Understand the working of Integer Multiplication? Designing the Multiplication algorithm using divide and conquer strategy? Analysis of the algorithm.
17. Define the terms associated with Decrease and Conquer strategy
18. Applying the concept of Decrease and Conquer to BFS? Analysis of the algorithm
19. Applying the concept of Decrease and Conquer to DFS? Analysis of the algorithm
20. Define the Transform and conquer strategy?
21. List the characteristics of Transform and conquer method?
22. Applying the concept of Transform and conquer to Presorting? Analysis of the algorithm.
23. Applying the concept of Transform and conquer to balanced search trees? Analysis of the algorithm
24. Applying the concept of Transform and conquer to Heap and Heap sort? Analysis of the algorithm
25. Define the Space and time tradeoffs strategy?
26. List the characteristics of Space and time tradeoffs method?
27. Applying the concept of Transform and conquer to sorting by counting? Analysis of the algorithm
28. Applying the concept of Transform and conquer to Input enhancement? Analysis of the algorithm
29. Explain Characteristics and Features of Problems Solved by Dynamic programming Algorithms.
30. Write the structure of Dynamic programming Algorithm.
31. Explain the all pair Shortest Path Problems? Identify the requirements of all pair Shortest Path Problems.
32. Give an Example for all pair Shortest Path Problem? Solve the Problems using Floyd's Algorithm? Derive the Complexity of the Algorithm.
33. Explain Knapsack Problems? Identify the requirements of Knapsack Problems.
34. Give an Example for Knapsack Problem? Solve the Problems using Dynamic programming.
35. Give an Example for all pair Shortest Path Problem? Solve the Problems using Floyd's Algorithm? Derive the Complexity of the Algorithm.
34. Explain the Spanning Tree Problems.
35. Identify the requirements of Minimum Spanning Tree Problems.
36. Give an Example for Minimum Spanning Tree Problem.

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 and analyze the machine instructions and program execution.	**L2, L4**	M	M	L	-	M	-	-	-	M	-	-	L	M
Understand and Explain the I/O organization	**L2**	L	-	-	-	L	-	-	-	M	L	-	M	M
Understand and explain the memory system.	**L2**	L	H	M	-	M	-	-	M	M	M	-	M	M
Apply the algorithms used for performing various arithmetic operations.	**L3**	H	M	L	-	-	-	-	-	M	M	-	M	M
Understand the execution of instruction with different bus architecture.	**L2**	-	L	L	-	-	-	-	L	M	L	-	L	M
Understand the operation of pipelining and multiprocessor.	**L4**	-	M	M	-	M	-	-	L	M	L	-	L	M

L- Low, M- Moderate, H-High

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 and analyze the machine instructions and program execution.	**L2, L4**	2	2	1	-	2	-	-	-	2	-	-	1	2
Understand and Explain the I/O organization	**L2**	1	-	-	-	1	-	-	-	2	1	-	2	2
Understand and explain the memory system.	**L2**	1	3	2	-	2	-	-	2	2	2	-	2	2
Apply the algorithms used for performing various arithmetic operations.	**L3**	3	2	1	-	-	-	-	-	2	2	-	2	2
Understand the execution of instruction with different bus architecture.	**L2**	-	1	1	-	-	-	-	1	2	1	-	1	2
Understand the operation of pipelining and multiprocessor.	**L4**	-	2	2	-	2	-	-	1	2	1	-	1	2

1 – Low, 2 – Moderate and 3 – High

Course Code : P13IS36	Semester : III	L - T - P : 3 - 0 - 1
Course Title : Object Oriented Programming and Java		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	**Weightage: CIE:50; SEE:50**	
Prerequisites : Computer Concepts and C Programming		

Course Learning Objectives (CLOs)

This course aims to

1. Explain the need of using Object Oriented programming in the real world applications.
2. Apply the concepts of data abstraction and data encapsulation
3. Write C++ programs using classes and objects.
4. Write Programs for automatic initialization of objects and destroy objects that are no longer required.
5. Construct applications to provide flexible options for the creation of new definitions for some of the operators.
6. Specifying mechanism of deriving new class from older classes through inheritance.
7. Implement methods to select appropriate member functions during run time.
8. Write a C++ program using templates and STL.
9. Handling exceptions.
10. Write Java programs using classes and objects.

Course Content

Unit – I

Introduction: A review of structures, Procedure Oriented programming systems, OOPS, Comparison of C++ with C, Console input/ Output in C++, variables in C++, Reference variable in C++, function prototyping, function overloading, Default values for formal arguments of functions, Inline functions.

Class and Objects: Introduction to Classes and Objects, Member functions and Member data, Objects and functions, Objects and arrays, Namespace.

10 Hrs

Unit – II

Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation.

Constructors and Destructors: Constructors, Destructors

Inheritance: Introduction to Inheritance, Base Class and Derived class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access Specifiers, Different Kinds of Inheritance.

10 Hrs

Unit – III

Virtual Functions: The Need for Virtual Functions, Virtual Functions, the Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors.

Operator Overloading: Operator Overloading, Overloading the Various Operators – Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Unary Minus and the Unary Plus Operator, Overloading the Arithmetic Operators. Overloading the Relational Operators, Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators

10 Hrs

8. Illustrate the Space and time Tradeoffs technique with respect to sorting by counting, input Enhancement in String Matching, hashing.L3
9. Analyze the Space and time Tradeoffs technique with respect to Sorting by Counting input Enhancement in String matching, hashing.L3
10. Examine whether space and Time Tradeoffs technique of algorithm design can be applied for a given problem and construct a space and time tradeoffs algorithms.L3

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

1. Explain Dynamic Programming method of algorithmic problem solving.L2
2. Illustrate the Dynamic programming technique with respect to Warshall's, and Knapsack problem.L3
3. Illustrate the Dynamic programming technique Floyd,s algorithm.L3
4. Analyze the dynamic Programming technique with respect to Warshall's, Floyd,s and Knapsack problem.L3
5. Examine whether Dynamic Programming technique of algorithm design can be applied for a given problem and construct a Dynamic programming algorithm.L2
6. Explain Greedy technique method of algorithmic problem solving.L2
7. Illustrate the Greedy technique with respect to prim's algorithm, Kruskal's Algorithm, Dijkstra's algorithm and Huffman trees.L3
8. Analyze the Greedy technique with respect to prim's algorithm, Kruskal's Algorithm, Dijkstra's algorithm and Huffman trees.L2
9. Examine whether Greedy technique of algorithm design can be applied for a given problem and construct a Greedy technique algorithm L3

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

1. Discuss the lower bound arguments.L1
2. Understanding the decision tree for searching pattern.L1
3. Understanding the decision tree for sorting patternL2
4. Discuss the P problems in backtracking L1
5. Discuss the NP problems in backtracking L3
6. Discuss the PNP complete problems. L2
7. Know the concept of Branch and Bound methodL1
8. Know the concept of for NP hard problems.L1

Review Questions

1. Identify the need of an Algorithm to solve a given Problem.
2. Write an Algorithm to solve the given real application Problem?
3. Analyze the space and Time Complexity of a given Algorithm.
4. What are the best case, worst case and average case time Complexities for insertion sort?
5. Define Ω, θ and O notations?
6. What are the factors involved in recursive and iterative algorithms?
7. How to identify whether the algorithm is recursive or iterative?
8. Define the terms associated with Brute Force strategy?
9. Applying the concept of Brute Force to Selection Sort? Analysis of the algorithm
10. Applying the concept of Brute Force to Bubble Sort? Analysis of the algorithm

Topic Learning Objectives

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

1. Definition of algorithm and discuss properties of an algorithm.L1
2. Explain the fundamental concept of algorithm problem solving. L2
3. Illustrate with example of some basic concept of graph terms. L5
4. Discuss the space complexity in algorithm design.L3
5. Discuss the Time Complexity in an algorithm Design. L3
6. Explain asymptotic notation and its properties. L2
7. Discuss the recursive algorithms.L3
8. Discuss the non-recursive algorithm.L3
9. Discuss some examples for designing algorithms.L3
10. Discuss and analyze some real world examples as a algorithm. L3

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

1. Explain brute Force algorithmic problem solving.L2
2. Illustrate the Brute force technique with respect to Selection Sort, Bubble Sort, Sequential Search and string Matching.L3
3. Examine whether brute force Technique of algorithm design can be applied for a given problem and construct a Brute force algorithm.L3
4. Explain Divide and Conquer method of algorithmic problem solving.L2
5. Illustrate the divide and conquer technique with respect to merge sort, quick sort, multiplication of large integers and strassen's matrix multiplication.L3
6. Analyze the Divide and Conquer algorithms with respect to Merge Sort, Quick Sort, multiplication of Large Integers and strassen's matrix Multiplication.L3
7. Explain Decrease and Conquer method of algorithmic problem solving.L2
8. Illustrate the decrease and conquer technique with respect to Breadth first Search, depth first Search.L3
9. Illustrate the decrease and conquer technique for topological Sort.L3
10. Analyze the Decrease and Conquer technique with respect to BFS, DFS and Topological Sort. L3

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

1. Explain Transform and conquer method of algorithmic problem Solving.L2
2. Illustrate the Transform and conquer technique for presorting, L3
3. Illustrate the Transform and conquer technique for balanced search Tree,L3
4. Illustrate the Transform and conquer technique for Heap and Heap Sort.L3
5. Analyze the Transform and conquer Technique With Respect to Presorting, Balanced Search tree, heap and heap Sort.L3
6. Examine whether Transform and Conquer technique of algorithm design can be applied for a given problem and construct a transform and conquer algorithm.L3
7. Explain space and time tradeoffs method of algorithmic problem solving.L2

Unit – IV

Stream Handling : Streams, The Class Hierarchy of Handling Streams, Opening and Closing Files, Files as Objects of the fstream Class, File Pointer, Random Access to Files

Templates: Introduction, Function Templates, Class Templates

Exception Handling: Introduction, C-Style Handling of Error generating Codes, C++ Style Solution – the try/throw/catch Construct, Limitation of Exception Handling.

10 Hrs

Unit – V

Introduction to JAVA: Java AS A PROGRAMMING TOOL, Features of Java, Two control statement, using blocks of code, Lexical Issues, The java class Libraries, Data Types, Variables & Arrays: the primitive types, integers, floating-point types, Characters, Booleans, Arrays operators and arithmetic operators, the bitwise operators, relational operators, Boolean logical operators, the assignment operators, operator precedence, using parentheses, control statements: java's selection statements, iteration statements, jump statements

Classes & Inheritance: Class Fundaments, Declaring Objects, Assigning Object Reference Variables, introducing methods, constructors, the finalize() Method, A stack class, overloading methods, using objects as parameters, argument passing, returning objects, recursion, introducing access control, understanding static, introducing final, Arrays revisited, introducing nested & inner classes, exploring the string class, using command-line arguments, Inheritance: using super, creating a multilevel Hierarchy, when constructors are called, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance

12 Hrs

Text Book:

1. "Object-Oriented Programming with C++", Sourav Sahay , Oxford University Press, 2006.
2. "The Complete Reference JAVA, J2SE", Herbert Schildt , 6th Edition, TMH, 2010.

Reference Books:

1. "C++ Primer ", Stanley B. Lippman, Josee Lajoie, Barbara E. Moo , 4th Edition, Addison Wesley, 2005.
2. "The Complete Reference C++", Herbert Schildt , 4th Edition, TMH, 2005.
3. "Programming with Java", Balaguruswamy, E , 4th Edition, 2010.

Course Outcomes

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

1. Distinguish between top-down and bottom-up programming approach and apply bottom-up approach to solve real world problems
2. Write programs for automatic initialization of objects and destroy objects that are no longer required.
3. Interpret the difference between static and dynamic binding. Apply both techniques to solve problems
4. Analyze generic data type for the data type independent programming which relates it to reusability.
5. Interpret and design the Exception Handling Techniques for resolving run -time errors and handle large data set using file I/O

Topic Learning Objectives

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

1. List the drawbacks of Structures.-L2
2. Compare C++ with C.-L3
3. Write a C++ program to perform console input and console output operations.-L2
4. Use of Reference variable in C++.-L2
5. Explain the need of function prototyping.-L2
6. Explain function overloading with an example.-L2
7. Illustrate the inline technique.-L2
8. Write the syntax of a class construct.-L2
9. Write a C++ program to implement Stack of integers.-L3
10. Write a C++ program to illustrate the use of arrays. –L2

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

1. Distinguish between Static memory management and Dynamic memory management.-L3
2. Explain the need of memory de-allocation.-L2
3. Write a C++ program to illustrate the use of Constructors.-L3
4. Write a C++ program to illustrate the use of Destructors.-L3
5. Design a C++ program to implement single level Inheritance.-L3
6. Illustrate the use of Base Class and Derived class Pointers.-L2
7. Define function overriding.-L1
8. List the advantages of Function Overriding.-L2
9. Write a C++ program to implement inheritance using the Protected Access Specifier.-L3
10. Compare different Kinds of Inheritance.-L3

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

1. Use of Virtual Functions.-L2
2. Explain the Mechanism of Virtual Functions with an example.-L2
3. Illustrate Pure virtual Functions.-L2
4. Explain the need of Virtual Destructors.-L2
5. Describe the order of invocation of Virtual Constructors.-L3
6. Explain the need of Operator Overloading.-L2
7. Write the syntax of an operator function.-L2
8. Write a program to demonstrate operator overloading using friend functions to overload ++ and –- operators.-L3
9. Write a program to demonstrate operator overloading using friend functions to overload arithmetic and relational operators.-L3
10. Write a program to demonstrate operator overloading using friend functions to overload insertion and extraction operators.-L3

Unit – III

Algorithm Design Methods-2:

Transform and conquer – introduction, Presorting, Balanced search trees, Heap and Heap sort, Space and time tradeoffs – sorting by counting, Input enhancement in string matching, Hashing, B-trees. Space and time tradeoffs – introduction, sorting by counting, sorting by distribution, Horspool's algorithm.

10 Hrs

Unit – IV

Algorithm Design methods-3:

Dynamic programming – introduction, Warshall's and Floyd's algorithms, the knapsack problem and memory functions, Greedy technique – Prim's algorithm, Kruskal's algorithm, Dijkstra's algorithm, Huffman's trees.

10 Hrs

Unit – V

Limitations of Algorithmic power:

Lower-bound arguments, Decision trees, P, NP, and NP-complete problems, Challenges of numerical algorithms, coping with the limitations of algorithm power – Backtracking, Branch and bound, Approximation algorithms for NP-hard problems, Algorithms for solving nonlinear equations.

10 Hrs

Text Books:

1. Anany Levitin: Introduction to the Design and Analysis of Algorithms, 2nd edition, Pearson Education, 2009.
2. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: Introduction to Algorithms, 3rd edition, PHI, 2011.

References:

1. E Horowitz, S Sahni, S Rajsekaran: Fundamentals of Computer Algorithms, 1st Edition, Galgotia Publication, 2012.
2. Richard Neopolitan, Kumarss Naimipour: Foundations of Algorithms using C++ pseudocode, 4th Edition, Jones & Bartlett India, 2011.
3. Sara Baase and Allen van Gelder: Computer algorithms - Introduction to Design and Analysis, 3rd Edition, Pearson Education, 2011.

Course Outcomes

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

1. **Analyze** the space and time complexities for the given problem.
2. **Solve** problems on searching and sorting using the algorithm techniques such as decrease and conquer, divide and conquer.
3. **Solve** graph based problems using the different algorithm techniques.
4. **Apply** solutions to overcome the limitations of algorithms
5. **Identify and Apply** algorithm Techniques to solve realistic problems.
6. **Design** and **implement** algorithms for the given problem.

Course Code : P13IS42	Semester : IV	L - T - P : 4 - 0 - 0
Course Title : **Analysis & Design of Algorithms**		
Contact Period: Lecture: 52 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50	
Prerequisites : Knowledge of Data Structure and C or C++ programming		

Course Learning Objectives (CLOs)

This course aims to

1. Explain the basic concepts of algorithm and its properties.
2. Explain the problem solving methods for designing algorithms
3. Explain the different problem solving concepts related to algorithm.
4. Explain the basic concepts of time complexity
5. Explain the basic concept of space complexity and various design strategies.
6. Describe the methodologies of how to analyze an algorithm based on divide and conquer method.
7. Describe the methodologies of how to analyze an algorithm based on decrease and conquer strategies.
8. Identify the complexity in the transform and conquer strategy and the input enhancement methods.
9. Apply and solve the ideas designed by dynamic programming method.
10. Apply and solve the ideas designed by dynamic programming greedy techniques
11. Explain the concept backtracking N P NP complete problems

Course Content

Unit – I

Introduction to Algorithms and Algorithm Efficiency:

What is an algorithm? Fundamentals of algorithmic problem solving, important problem types, Fundamental data structures, Fundamentals of the analysis of algorithm efficiency – Analysis framework, asymptotic notations and basic efficiency classes, Mathematical analysis of recursive and non recursive algorithms*.*

10Hrs

Unit – II

Algorithm Design Methods-1:

Brute force – introduction, selection sort and bubble sort, Sequential search and brute-force string matching. Divide and conquer – introduction, Merge sort, Quick sort, Binary search, Binary tree traversals and related properties, Decrease and conquer – introduction, insertion sort, depth-first and breadth-first search, Topological sorting.

12Hrs

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

1. Define Stream.-L1
2. Depict the Class Hierarchy of Handling Streams.-L2
3. Write a C++ program to illustrate Opening and Closing Files.-L3
4. Write a C++ program to illustrate random access to Files.-L3
5. Use of Templates.-L2
6. Write a C++ program to illustrate the function template.-L3
7. Write a C++ program to illustrate the class template.-L3
8. Explain the need of handling exceptions.-L2
9. Write a C++ program to handle exceptions.-L3
10. List the Limitations of Exception Handling.-L2

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

1. Explain the features of Java.-L2
2. Write a Java program to illustrate the use of data types, variables and Arrays.-L3
3. Write a Java program to illustrate the use of different operators.-L3
4. Write a Java program to illustrate the use of control statements.-L3
5. Write a Java program to illustrate the creation and use of objects.-L3
6. Explain the role of finalize() method.-L2
7. List the uses of Nested classes.-L2
8. Write a java program to illustrate dynamic method dispatch.-L3
9. Write a java program to illustrate the use of abstract classes.-L3
10. Explain the need of final keyword in Java.-L2

Review Questions

1. Define object oriented programming.
2. Explain the features of OOP.
3. Differentiate object oriented programming and procedure oriented programming.
4. Define function overloading.
5. What is a class? How does it accomplish data hiding?
6. How is a member function of a class defined?
7. Distinguish between private vs. public member functions.
8. Explain friend classes and friend functions.
9. Define friend function in C++. Write a C++ program with use of object as function arguments.
10. When do we declare a member of a class static?
11. What are the merits and demerits of using friend functions?
12. Explain new and delete expressions with examples.
13. What is a constructor? Explain different types of constructors.
14. Describe the importance of destructors.
15. Differentiate between constructors and destructors.
16. How do we invoke a constructor function?
17. Write a program to demonstrate: i. array of objects ii. pointers to objects
18. Write a program to demonstrate: i. this pointer ii. pointers to class members
19. Write a program to demonstrate: i. pointers to derived types ii. Copy constructor
20. What is inheritance? Explain with an example.
21. What are the different forms of inheritance? Give an example for each.
22. Write a program for multiple inheritances.
23. What is a virtual base class?
24. When do we make a class virtual?
25. What is an abstract class?
26. Write a program to demonstrate the calling a virtual function trough a base class reference
27. Write a program to demonstrate Pure virtual functions
28. Explain early and late binding.
29. How is polymorphism achieved at:
    (i) compile time
    (ii) run time
30. When do we make a virtual function “pure”?
31. What are the implications of making a function a pure virtual function?
32. What is operator overloading? Why is it necessary to overload an operator?
33. What is an operator function? Describe the syntax of an operator function.
34. Briefly describe the class hierarchy provided by C++ for stream handling.
35. Describe how the contents of a disk file can be randomly accessed in C++.

Course Code : P13ISL38	**Semester : III**	**L - T - P : 0 - 0 - 1.5**
Course Title : Digital Design Lab		
Contact Period: Lecture: 36 Hr, Exam: 3 Hr		**Weightage: CIE:50; SEE:50**
Prerequisites : Digital Design		

Course Learning Objectives (CLOs)

This course aims to

1. Design and implement different combinational circuits.
2. Design and implement different sequential circuits.
3. Design and implement D/A converter.

Contents

PART A

1. Design a circuit for Full Adder.
2. Design a circuit for code Conversion.
3. Application of MUX/DEMUX.
4. Application of Decoder.
5. Design of 3 bit Synchronous Counter.
6. Design of Asynchronous Counter.
7. Design of Ring Counter/Johnson Counter.
8. Design of Sequence generator/detector.
9. Digital to analog Converter.

PART B

1. Write the Verilog/VHDL code for a Full Adder. Simulate and verify its working.
2. Write the Verilog/VHDL code for 8:1 MUX. Simulate and verify its working.
3. Write the Verilog/VHDL code for a 3:8 decoder. Simulate and verify its working.
4. Write the Verilog/VHDL code for a Flip-Flop with positive-edge triggering. Simulate and verify its working.
5. Write the Verilog/VHDL code for a mod-8 up counter. Simulate and verify its working.
6. Write the Verilog/VHDL code for a Ring Counter. Simulate and verify its working.
7. Write the Verilog/VHDL code for a Johnson Counter. Simulate and verify its working.

Note: In SEE, student has to pick a lot for question that contains subsections from PART A and PART B and has to execute both the subsections compulsorily.

Course Code : P13ISL37	Semester : III	L - T - P : 0 - 0 - 1.5
Course Title : Data structures Lab		
Contact Period: Lecture: 36 Hr, Exam: 3 Hr	Weightage: CIE:50; SEE:50	
Prerequisites : Computer Concepts and C Programming		

Course Learning Objectives (CLOs)

This course aims to

1. Apply different concepts of data structures to solve real time problems.
2. Distinguish between iterative method and recursive method.
3. Apply the concept of recursion, stack, queues and Linked list to solve various applications.
4. Solve non-linear data structures, such as binary tree.
5. Implement different sorting and searching techniques.

List of Programs

1. Write the C programs using Recursion.
2. Write a C program to implement the stack of integers. The program should print appropriate messages for stack overflow, stack underflow and stack empty.
3. Write a C program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression.
4. Write a C program to evaluate a valid suffix/postfix expression using stack.
5. Write a C program using dynamic variables and pointers, to implement a singly linked list.
6. Write a C program to implement the applications of single linked list.
7. Write a C program to implement Queue of integers using an array.
8. Write a C program to implement circular Queue of integers using an array.
9. Write a C program using dynamic variables and pointers to construct a queue of integers using singly linked list.
10. Write a C program to simulate the working of a Double Ended Queue of integers using an array.
11. Write a C program to design a priority queue.
12. Write a C program to demonstrate the working of binary search tree.
13. Write a C program to sort the given list of N numbers.
14. Write a program to search an element in a given list of N numbers.

36. What is the need for function templates? How are they created?
37. When and how does the C++ compiler generate an actual function definition from its template?
38. What is the Standard Template Library?
39. What is the need for Exception handling?
40. How are uncaught exceptions caught?
41. What is the limitation of exception handling in C++?
42. Explain the features of Java.
43. List the advantages of Java.
44. Which are the Similarities and Differences between Java and C++?
45. Explain the need of inner class with an example.
46. Write a Java program to illustrate the use of super keyword in Java.
47. Write a Java program to illustrate dynamic method dispatch in Java.
48. Write a Java program to implement simple calculator.
49. Write a Java program using classes and objects.
50. List out the control statements available in Java.

Lesson Plan

Unit – I

1. A review of structures
2. Procedure Oriented programming systems, OOPS
3. Comparison of C++ with C, Console input/ Output in C++
4. variables in C++, Reference variable in C++
5. function prototyping, function overloading
6. Default values for formal arguments of functions, Inline functions
7. Introduction to Classes and Objects
8. Member functions and Member data
9. Objects and functions
10. Objects and arrays , Namespace

Unit – II

1. Introduction to Dynamic Memory Management
2. Dynamic Memory Allocation, Dynamic Memory Deallocation
3. Constructors
4. Different types of Constructors
5. Destructors
6. Introduction to Inheritance
7. Base Class and Derived class Pointers
8. Function Overriding, Base Class Initialization
9. The Protected Access Specifier, Deriving by Different Access Specifiers
10. Different Kinds of Inheritance

Unit – III

1. The Need for Virtual Functions, Virtual Functions
2. The Mechanism of Virtual Functions, Pure Virtual Functions
3. Virtual Destructors and Virtual Constructors
4. Operator Overloading
5. Overloading the Increment and the Decrement Operators(Prefix, postfix form)
6. Overloading the Unary Minus and the Unary Plus Operator
7. Overloading the Arithmetic Operators, Relational Operators
8. Overloading the Assignment Operator
9. Overloading the Insertion and Extraction Operators
10. Examples

Unit – IV

1. Streams, The Class Hierarchy of Handling Streams
2. Opening and Closing Files, Files as Objects of the fstream Class
3. File Pointer, Random Access to Files
4. Templates: Introduction
5. Function Templates
6. Class Templates
7. Exception Handling: Introduction
8. C-Style Handling of Error generating Codes
9. C++ Style Solution – the try/throw/catch Construct
10. Limitation of Exception Handling.

Unit – V

1. Introduction to JAVA: Java AS A PROGRAMMING TOOL, Features of Java, Two control statement
2. Using blocks of code, Lexical Issues, The java class Libraries
3. Data Types, Variables & Arrays: the primitive types, integers, floating-point types, Characters, Booleans, Arrays
4. Operators and arithmetic operators, the bitwise operators, relational operators, Boolean logical operators, the assignment operators, operator precedence, using parentheses
5. Control statements: Java's selection statements, iteration statements, jump statements
6. Class Fundaments, Declaring Objects, Assigning Object Reference Variables, introducing methods, constructors, the finalize() Method
7. A stack class, overloading methods, using objects as parameters, argument passing, returning objects, recursion
8. Introducing access control, understanding static, introducing final, Arrays revisited
9. Introducing nested & inner classes, exploring the string class
10. Using command-line arguments, Inheritance: using super
11. Creating a multilevel Hierarchy, when constructors are called, method overriding, dynamic method dispatch
12. Using abstract classes, using final with inheritance

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
Distinguish between top-down and bottom-up programming approach and apply bottom-up approach to solve real world problems	L3	M	M	H	M	H	-	-	-	H	-	-	H	H
Write programs for automatic initialization of objects and destroy objects that are no longer required.	L3	-	M	-	-	L	-	-	-	H	-	-	-	M
Interpret the difference between static and dynamic binding. Apply both techniques to solve problems	L3	-	M	L	M	M	-	-	-	H	-	-	M	M
Analyze generic data type for the data type independent programming which relates it to reusability.	L4	-	M	M	-	-	-	-	-	H	-	-	M	M
Interpret and design the Exception Handling Techniques for resolving run-time errors and handle large data set using file I/O	L3	-	M	M	-	-	-	-	-	H	-	-	M	M

L- Low, M- Moderate, H-High

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
Distinguish between top-down and bottom-up programming approach and apply bottom-up approach to solve real world problems	L3	2	2	3	2	3	-	-	-	3	-	-	3	3
Write programs for automatic initialization of objects and destroy objects that are no longer required.	L3	-	2	-	-	1	-	-	-	3	-	-	-	2
Interpret the difference between static and dynamic binding. Apply both techniques to solve problems	L3	-	2	1	2	2	-	-	-	3	-	-	2	2
Analyze generic data type for the data type independent programming which relates it to reusability.	L4	-	2	2	-	-	-	-	-	3	-	-	2	2
Interpret and design the Exception Handling Techniques for resolving run-time errors and handle large data set using file I/O	L3	-	2	2	-	-	-	-	-	3	-	-	2	2

1 – Low, 2 – Moderate and 3 – High