MVJ COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

III SEMESTER COURSE DIARY

INDEX

ELECTRONIC CIRCUITS

10CS32

ELECTRONIC CIRCUITS
(Common to CSE & ISE)

Subject Code: 10CS32	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART - A

UNIT - 1 7 Hours

Transistors, UJTs, and Thyristors: Operating Point, Common-Emitter Configuration, Thermal Runaway, Transistor Switch, Unijunction Transistors, SCR.

UNIT - 2 6 Hours

Field Effect Transistors: Bipolar Junction Transistors versus Field Effect Transistors, Junction Field Effect Transistors, Metal Oxide Field Effect Transistors, Differences between JFETs and MOSFETs, Handling MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices, Insulated Gate Bipolar Transistors (IGBTs)

UNIT - 3 6 Hours

Optoelectronic Devices: Introduction, Photosensors, Photoconductors, Photodiodes, Phototransistors, Light-Emitting Diodes, Liquid Crystal Displays, Cathode Ray Tube Displays, Emerging Display Technologies, Optocouplers

UNIT - 4 7 Hours

Small Signal Analysis of Amplifiers: Amplifier Bandwidth: General Frequency Considerations, Hybrid h-Parameter Model for an Amplifier, Transistor Hybrid Model, Analysis of a Transistor Amplifier using complete h-Parameter Model, Analysis of a Transistor Amplifier Configurations using Simplified h-Parameter Model (CE configuration only), Small-Signal

Analysis of FET Amplifiers, Cascading Amplifiers, Darlington Amplifier, Low-Frequency Response of Amplifiers (BJT amplifiers only).

PART - B

UNIT - 5 6 Hours

Large Signal Amplifiers, Feedback Amplifier: Classification and characteristics of Large Signal Amplifiers, Feedback Amplifiers: Classification of Amplifiers, Amplifier with Negative Feedback, Advantages of Negative Feedback, Feedback Topologies, Voltage-Series (Series-Shunt) Feedback, Voltage-Shunt (Shunt-Shunt) Feedback, Current-Series (Series-Series) Feedback, Current-Shunt (Shunt-Series) Feedback.

UNIT - 6 7 Hours

Sinusoidal Oscillators, Wave-Shaping Circuits: Classification of Oscillators, Conditions for Oscillations: Barkhausen Criterion, Types of Oscillators, Crystal Oscillator, Voltage-Controlled Oscillators, Frequency Stability.

Wave-Shaping Circuits: Basic RC Low-Pass Circuit, RC Low-Pass Circuit as Integrator, Basic RC High-Pass Circuit, RC High-Pass Circuit as Differentiator, Multivibrators, Integrated Circuit (IC) Multivibrators.

UNIT - 7 7 Hours

Linear Power Supplies, Switched mode Power Supplies: Linear Power Supplies: Constituents of a Linear Power Supply, Designing Mains Transformer; Linear IC Voltage Regulators, Regulated Power Supply Parameters.

Switched Mode Power Supplies: Switched Mode Power Supplies, Switching Regulators, Connecting Power Converters in Series, Connecting Power Converters in Parallel

UNIT - 8 6 Hours

Operational Amplifiers: Ideal Opamp versus Practical Opamp, Performance Parameters, Some Applications: Peak Detector Circuit, Absolute Value Circuit, Comparator, Active Filters, Phase Shifters, Instrumentation Amplifier, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter, Sine Wave Oscillators.

Text Book:

1. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2009.
(4.1, 4.2, 4.7, 4.8, 5.1 to 5.3, 5.5, 5.6, 5.8, 5.9, 5.13, 5.14, 6.1, 6.3, 7.1 to 7.5, 7.10 to 7.14, Listed topics only from 8, 10.1, 11, 12.1, 12.2, 12.3, 12.5, 13.1 to 13.6, 13.9, 13.10, 14.1, 14.2, 14.6, 14.7, 15.1, 15.5 to 15.7. 16.3, 16.4, 17.12 to 17.22)

M.V.J. College of Engineering
Department of Computer Science& Engineering

LESSON PLAN

Class : III Sem **Total Hours: 62**
Subject: Electronic Circuits **Hours week: 5**
Subject Code: 06CS32 **IA Marks : 25**

Sl. No.	Chapter	Hour No	Topics to be covered
1	Transistors, UJTs, and Thyristors	1	Operating Point
		2	Common-Emitter Configuration
		3	Thermal Runaway
		4	Transistor Switch
		5	Unijunction Transistors
		6	SCR
		7	SCR
		8	SCR
2	Field Effect Transistors	9	Bipolar Junction Transistors versus Field Effect Transistors
		10	Junction Field Effect Transistors
		11	Metal Oxide Field Effect Transistors
		12	Differences between JFETs and MOSFETs
		13	Handling MOSFETs
		14	Biasing MOSFETs, FET Applications
		15	CMOS Devices
		16	Insulated Gate Bipolar Transistors (IGBTs)
3	Optoelectronic Devices	17	Introduction, Photosensors
		18	Photoconductors, Photodiodes

		19	Phototransistors, Light-Emitting Diodes
		20	Liquid Crystal Displays
		21	Cathode Ray Tube Displays
		22	Emerging Display Technologies
		23	Optocouplers
		24	Optocouplers
		25	Revision
4	Small Signal Analysis of Amplifiers	26	Amplifier Bandwidth: General Frequency Considerations
		27	Hybrid h-Parameter Model for an Amplifier
		28	Transistor Hybrid Model
		29	Analysis of a Transistor Amplifier using complete h-Parameter Mode
		30	Analysis of a Transistor Amplifier Configurations using Simplified h-Parameter Model (CE configuration only
		31	Small-Signal Analysis of FET Amplifiers
		32	Cascading Amplifiers
		33	Darlington Amplifier, Low-Frequency Response of Amplifiers (BJT amplifiers only).
5	Large Signal Amplifiers, Feedback Amplifier	34	Classification and characteristics of Large Signal Amplifiers
		35	Feedback Amplifiers
		36	Classification of Amplifiers
		37	Amplifier with Negative Feedback
		38	Advantages of Negative Feedback, Feedback Topologies

		39	Voltage-Series (Series-Shunt) Feedback, Voltage-Shunt (Shunt-Shunt) Feedback
		40	Current-Series (Series-Series) Feedback, Current-Shunt (Shunt-Series) Feedback.
6	Sinusoidal Oscillators, Wave-Shaping Circuits	41	Classification of Oscillators
		42	Conditions for Oscillations: Barkhausen Criterion
		43	Types of Oscillators, Crystal Oscillator
		44	Voltage-Controlled Oscillators, Frequency Stability.
		45	Wave-Shaping Circuits: Basic RC Low-Pass Circuit
		46	RC Low-Pass Circuit as Integrator, Basic RC High-Pass Circuit
		47	RC High-Pass Circuit as Differentiator, Multivibrators
		48	Integrated Circuit (IC) Multivibrators.
7	Linear Power Supplies, Switched mode Power Supplies	49	Linear Power Supplies
		50	Constituents of a Linear Power Supply
		51	Designing Mains Transformer
		52	Linear IC Voltage Regulators
		53	Regulated Power Supply Parameters.
		54	Switched Mode Power Supplies: Switched Mode Power Supplies
		55	Switching Regulators, Connecting Power Converters in Series
		56	Connecting Power Converters in Parallel
8	Operational Amplifiers	57	Ideal Opamp versus Practical Opamp
		58	Performance Parameters, Some Applications: Peak Detector Circuit
		59	Absolute Value Circuit, Comparator, Active Filters

		60	Phase Shifters, Instrumentation Amplifier, Non-Linear Amplifier
		61	Relaxation Oscillator, Current-To-Voltage Converter
		62	Voltage-To-Current Converter, Sine Wave Oscillators

Signature of HOD **Signature of Staff**

ELECTRONIC CIRCUIT
QUESTION BANK

Sub Code: 10CS32
Hour/week: 04
Total hours: 62

---

1) Briefly explain clippers and types of clippers?
2) Give shorts on the following
   i) Pin code ii) current Regulator diodes. iii) varistors
3) List the opto-electronic devices and explain
4) Draw the symbol for varactor diode, its operation.
5) Draw and explain Base biased amplifier?
6) What is need of small signal operation?
7) Briefly explain how will you analyze the following circuit
   i) Emitter based amplifier
8) Give definition for the following
   i) Voltage gain. ii) Current gain iii) Ac Beta.
9) Give the condition for coupling capacitor and Bypass capacitor?
10) What is the use of parameters? Types?
11) Device the expression for voltage amplifier
12) What you meant by multistage amplifier and how can you differentiate single stage amplifier?
13) How will you troubleshoot problems for the following amplifier
   i) Multistage amplifier ii) swamped amplifier
14) Find the o/p impedance of cc amplifier? Explain the
15) What do you mean by cascading ad cascading CC & CE amplifiers.
16) Explain voltage regulation?
17) Give the difference between for Amplifier terms?
18) Explain the difference between class A and class B amplifier
19) Give all formulas of class C amplifier
20) Briefly explain the basing of class B/AB amplifiers.
21) Write short note on the following
   i) Two land lines ii) Transistor power Rating.
22) Draw the circuit for class B push pull emitter follower and explain its operation?
23) Explain in detail about class B/AB drives
24) Define:
   i) Depletion mode ii) Enhancement mode
25) Explain ohmic region in MOSFATS.
26) What is MOSFAT and types?
27) Write short note in digital switching.
28) Write short notes o CMOS technology?
29) give the example for bode pot
30) define: (1) decibel power gain
   (2) decibel voltage gain

31)define feedback & types of feedback?
32)derive the equations of VCVS?
33.explain the operation of ICVS?
34.differentiate ICVS and VCVS
35.write short notes on the following
i.bandwidth
ii.decibels above a reference
36.what is comparator?with zero ref and non zero ref
37.generate a square wave from sinewave?
38.explain the operation of 555 timer
39.explain about other 555 timer
40)list the function generator Ics
41) What is multivibrator? Explain the operation of astable multivibrator of 555 timer?
42.define regulators? Types of regulators?
43) What are the characteristics of supply?
44) Explain the operation of current boosters?
45) Write short notes on DC to Dc converters?
46) Briefy explain the operation of Dc to DC convertor?
47) List the advantages of switching regulators?
48) Write short notes on switching regulators and monolithic regulators?
49) Define hysterisis and expain the operation of comparators with hysterisis?
50) Explain (i) phase locked loop
(ii)limiters

LOGIC DESIGN

10CS33

M.V.J. COLLEGE OF ENGINEERING

Department of Computer Science & Engineering

SYLLABUS

LOGIC DESIGN

(Common to CSE & ISE)

Sub. Code: 10CS33 IA Marks: 25
Hrs/Week: 04 Exam Hours: 03
Total Hours: 52 Exam Marks: 100

PART A

Unit I **7 Hours**

***Digital Principles, Digital Logic*:** Definitions for Digital Signals, Digital Waveforms, Digital Logic, 7400 TTL Series, TTL Parameters The Basic Gates: NOT, OR, AND, Universal Logic Gates: NOR, NAND, Positive and Negative Logic, Introduction to HDL.

Unit II **6 Hours**

***Combinational Logic Circuits*:** Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard Covers, HDL Implementation Models .

Unit III **6 Hours**

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, Encoders, Exclusive-or Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits.

Unit IV **7 Hours**

Clocks , Flip-Flops: Clock Waveforms, TTL Clock, Schmitt Trigger, Clocked D FLIP-FLOP, Edge-triggered D FLIP-FLOP, Edge-triggered JK FLIP-FLOP, FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, Analysis of Sequential Circuits, HDL Implementation of FLIP-FLOP .

PART B

Unit V **6 Hours**

Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register Implementation in HDL.

Unit VI **7 Hours**

Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus, Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL .

Unit VII **7 Hours**

Design of Synchronous and Asynchronous Sequential Circuits: Design of Synchronous Sequential Circuit: Model Selection, State Transition Diagram, State Synthesis Table, Design Equations and Circuit Diagram, Implementation using Read Only Memory, Algorithmic State Machine, State Reduction Technique.
Asynchronous Sequential Circuit: Analysis of Asynchronous Sequential Circuit, Problems with Asynchronous Sequential Circuits, Design of Asynchronous Sequential Circuit, FSM Implementation in HDL.

Unit VIII **6 Hours**

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

Text Book:

1. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 7th Edition, Tata McGraw Hill, 2010.

Reference Books:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition, Tata McGraw Hill, 2005.
2. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
3. Charles
4. H. Roth: Fundamentals of Logic Design, Jr., 5th Edition, Cengage Learning, 2004.
5. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss: Digital Systems Principles and Applications, 10th Edition, Pearson Education, 2007.
6. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson Education, 2008.

M..V.J. COLLEGE OF ENGINEERING

Department of Computer Science & Engineering

LOGIC DESIGN

LESSON PLAN

Class : III SEM **Total Hours: 62**
Subject: Logic Design **Hours/week: 5**
Subject Code: 10CS33 **IA Marks : 25**

Sl. No.	Unit	Hour No	Topics to be covered
1	Digital Principles, Digital Logic	1.	Definitions for Digital Signals, Digital Waveforms
		2.	Digital Logic, 7400 TTL Series, TTL Parameters
		3.	Basic Gates: NOT, OR, AND
		4.	Basic Gates: NOT, OR, AND
		5.	Universal Logic Gates: NOR, NAND
		6.	Positive and Negative Logic
		7.	Introduction to HDL
		8.	Revision & Solving Exercise Problems
2	Combinational Logic Circuits	9.	Introduction to Boolean Laws & Theorems , Sum-of-Products Method
		10.	Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications
		11.	Don't-care Conditions ,Product-of-sums Method, Product-of-sums simplifications
		12.	Simplification by Quine-McClusky Method
		13.	Simplification by Quine-McClusky Method
		14.	Hazards and Hazard Covers, HDL Implementation Models.
		15.	Revision & Solving Exercise Problems
3	Data processing circuits	16.	Multiplexers, Demultilexers
		17.	1-of-16 decoder, Ex-or gates
		18.	Parity generators and checkers
		19.	Magnitude comparator
		20.	Programmable array logic, Pogrammable logic array
		21.	HDL Implementations of data Processing Circuit
		22.	Revision & Solving Exercise Problems
4	Clocks,Flip- flops	23.	Clock waveforms
		24.	Clocked D FLIP-FLOP, Edge-triggered D FLIP-FLOP,
		25.	Edge-triggered JK FLIP-FLOP
		26.	FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits
		27.	Various Representation of FLIP-FLOPs, Analysis of Sequential Circuits
		28.	HDL Implementation of FLIP-FLOP.
		29.	HDL Implementation of FLIP-FLOP.

		30.	Revision & Solving Exercise Problems
5	Registers	31.	Types of registers, Serial in serial out,
		32.	Serial in parallel out
		33.	Parallel in Serial out
		34.	Parallel in parallel out
		35.	Universal Shift Registers, Applications of shift registers.
		36.	Register implementation in HDL
		37.	Revision & Solving Exercise Problems
6	Counters	38.	Asynchronous counters, Decoding gates
		39.	Synchronous counters, changing the counter modulus
		40.	Decade counters
		41.	Presettable counters
		42.	Presettable counters
		43.	Counter design as a synthesis problem
		44.	A digital clock, Counter design using HDL
		45.	Revision & Solving Exercise Problems
7	Design of Synchronous and Asynchronous sequential circuit	46.	Model selection, State transition diagram
		47.	State synthesis table, Design equations and circuit diagram
		48.	Implementation using read only memory
		49.	Algorithmic state machine, State reduction technique
		50.	Analysis of asynchronous sequential circuit
		51.	Problems with asynchronous sequential circuits
		52.	Design of asynchronous sequential circuits, FSM Implementation in HDL
		53.	Revision & Solving Exercise Problems
8	D/A conversion and A/D conversion	54.	Variable, resistor networks, binary ladders
		55.	D/A converters, D/A accuracy and resolution
		56.	A/D converter – simultaneous conversion
		57.	A/D converter-counter method
		58.	Continuous A/D conversion, A/D techniques
		59.	Dual slope A/D conversion, A/D accuracy and resolution
		60.	Revision & Solving Exercise Problems
		61.	Solving University Questions
		62.	Solving University Questions

Signature of Staff **Signature of HOD**

M.V.J. College of Engineering
Department of Computer Science& Engineering

Logic Design

QUESTION BANK

1) Implement the logic circuit that has the expression $x=\overline{AB.\ \overline{(c+d)}}$ using only NOR and NAND gates. Determine the output level for the same when A=B=C=1 and D=0.
2) Design a logic circuit with inputs P, Q, R, so that o/p is high whenever P is 0 or whenever Q=R=1.
3) Design a logic Circuit that has three inputs A, B&C and whose o/p will be high only when a majority of the i/p is high.
4) What are universal gates? Explain them.
5) What is a Karnaugh map? Reduce the following function using k-map technique and Implement using gates.
   i) $F(P,Q,R,S)= \sum m(0,1,4,8,9,10)+d(2,11)$.
   ii) $F(A,B,C,D)=ABD+ABCD+ABD+ABCD$.
6) Simplify the following 5 various Boolean expression using Quine-Mccluskey method $f=\sum m(10,1,9,15,24,29,30)+d(8,11,31)$
7) Using quine – McCluskey method & prime implicant reduction, determine the minimal SOP expression for the following $f(w,x,y,z)= M(0,4,5,9). D(1,7,13)$
8) Explain the Ex- or function.
9) What is Decoder? Using gates, show how do you design 3 to 8 line decoder.
10) Explain different HDL implementation models
11) Illustrate the use of PLA for combinational logic design with an example.
12) What are glitches? Explain in detail.
13) What is the race around condition in flip flop? How it can be eliminated?
14) State various uses of shift registers
15) What are registers and how many types of registers were there? Explain serial in serial out registers.
16) Draw the block diagram of a mod-7 twisted ring counter and explain its operation. Give the count sequence table and the decoding logic used to identify the various states.
17) Design a serial adder using sequential logic procedure.
18) Design a combinational circuit which converts from 4-bit binary number to 2's complement o/p.
19) What are state reduction techniques? Explain.
20) What are problems with asynchronous sequential circuits?
21) Discuss how excitation tables, state tables and state diagrams are used to analyse a synchronous sequential network.
22) Explain A/D converter using counter method.
23) With the aid of a neat circuit diagram explain the operation of a 2 input TTL nand gate with totem o/p.
24) How does a parallel counter differ from a serial counter?
25) Show a method for constructing a 5*2(mod 10) decade counter.

26) The total propagation delay through a 74HC04 inverter is known to be 24ns. What is the maximum clock frequency that can be used with this device?
27) What is the main difference between an edge triggered and pulse triggered JK flip-flop?
28) How is excitation table different from flip-flop truth table?
29) What is the primary cause of glitches that sometimes occur at the o/p of a decoding gate used with a ripple counter? What is one method to eliminate these glitches?
30) Explain different models by which a synchronous sequential logic circuit can be designed?
31) Explain 4 bit shift register in detail and give its timing diagram.

DATA STRUCTURES WITH C
10CS35

Data Structures with C
Syllabus

Sub. Code : 10CS35 **IA Marks : 25**
Hrs/Week : 04 **Exam Hours : 03**
Total Hours : 52 **Exam Marks : 100**

Part A

Unit I
BASIC CONCEPTS: Pointers and Dynamic Memory Allocation, Algorithm Specification, Data Abstraction, Performance Analysis, Performance Measurement **8 Hours**

Unit II
ARRAYS and STRUCTURES: Arrays, Dynamically Allocated Arrays, Structures and Unions, Polynomials, Sparse Matrices, Representation of Multidimensional Arrays **6 Hours**

Unit III
STACKS AND QUEUES: Stacks, Stacks Using Dynamic Arrays, Queues, Circular Queues Using Dynamic Arrays, Evaluation of Expressions, Multiple Stacks and Queues. **6 Hours**

Unit IV
LINKED LISTS: Singly Linked lists and Chains, Representing Chains in C, Linked Stacks and Queues, Polynomials, Additional List operations, Sparse Matrices, Doubly Linked Lists **6 Hours**

Part B

Unit V
TREES – 1: Introduction, Binary Trees, Binary Tree Traversals, Threaded Binary Trees, Heaps. **6 Hours**

Unit VI
TREES – 2, GRAPHS: Binary Search Trees, Selection Trees, Forests, Representation of Disjoint Sets, Counting Binary Trees, The Graph Abstract Data Type. **6 Hours**

Unit VII
PRIORITY QUEUES Single- and Double-Ended Priority Queues, Leftist Trees, Binomial Heaps, Fibonacci Heaps, Pairing Heaps. **6 Hours**

Unit VIII

EFFICIENT BINARY SEARCH TREES: Optimal Binary Search Trees, AVL Trees, Red-Black Trees, Splay Trees. **7 Hours**

Text Books

1. Horowitz, Sahni, Anderson-Freed: Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2007.
(Chapters 1, 2.1 to 2.6, 3, 4, 5.1 to 5.3, 5.5 to 5.11, 6.1, 9.1 to 9.5, 10)

Reference Books

1. Yedidyah, Augenstein, Tannenbaum: Data Structures Using C and C++, 2nd Edition, Pearson Education, 2003.
2. Debasis Samanta: Classic Data Structures, 2nd Edition, PHI, 2009.
3. Richard F. Gilberg and Behrouz A. Forouzan: Data Structures A Pseudocode Approach with C, Cengage Learning, 2005.

M.V. J. College of Engineering

Department of Computer Science & Engineering

DATA STRUCTURES WITH C

Lesson Plan

Subject Code: 10CS35 **I.A. Marks: 25**

Hours/Week: 04 Exam Hours: 03

Total Hours: 52[50min*62classes] **Exam Marks: 100**

S.No	Chapter	Hours	Topics to be covered
1.	**Basic Concepts**	4	Pointer and Dynamic Memory Allocation
		5	Algorithm Specification
		6	Data Abstraction
		7	Performance Analysis
		8	Performance Measurement
2.	**Arrays and Structures**	9	Arrays, Dynamically Allocated Arrays
		10	Structures
		11	Unions
		12	Polynomials
		13	Sparse Matrices
		14	Representation of Multidimensional Arrays
3	**Stacks and Queues**	15	Stacks , Stacks using Dynamic Arrays
		16	Stacks using Dynamic Arrays Queues
		17	Queues Circular Queues using Dynamic Arrays
		18	Circular Queues using Dynamic Arrays
		19	Evaluation of Expression
		20	Multiple Stacks and Queues
4.	**Linked Lists**	21	Singly Linked Lists and Chains
		22	Representing Chains in C
		23	Linked Stacks and Queues
		24	Polynomials. Additional List Operations
		25	Sparse Matrices
		26	Doubly Linked Lists
5.	**Trees 1**	27	Introduction
		28	Binary Trees
		29	Binary Tree Traversals
		30	Binary Tree Traversals
		31	Threaded Binary Trees
		32	Heaps
6.	**Tree 2 Graphs**	33	Binary Search Trees

		34	Selection Trees
		35	Forests
		36	Representation of disjoint Sets
		37	Counting Trees
		38	The Graph Abstract Data Type
7.	**Priority Queues**	39	Single and Double ended Priority Queues
		40	Single and Double ended Priority Queues
		41	Leftist Trees
		42	Binomial Heaps
		43	Fibonacci Heaps
		44	Pairing Heaps
8.	Efficient Binary Search Tree	45	Optimal Binary Search Trees
		46	Optimal Binary Search Trees
		47	AVL Trees
		48	AVL Trees
		49	Red Black Tress
		50	Red Black Tress
		51	Splay Trees
		52	Splay Trees

Signature of HOD **Signature of Staff**

MVJ COLLEGE OF ENGINEERING
BANGALORE – 560 078

Department of Computer Science and Engineering

Question Bank

Class : III Sem Hours / week : 04

Subject: Data Structures with C Sub Code : 10CS35

1. What is pointer?
2. How do you declare a pointer?
3. How do you initialize the pointer?
4. What is meant by pointer to function?
5. List the memory allocation functions available in C.
6. How do you relate array with a pointer with example?
7. Define String.
8. How do you initialize the Strings?
9. List of memory IO functions available.
10. Differentiate formatted and unformatted IO functions with respect to String.
11. What are the different operations can be performed on String?
12. List out some string manipulation functions.
13. Define structure.
14. How do you initialize the structure members?
15. How to access the access members?
16. Define Union.
17. Differentiate Union and Structures.
18. How do you pass structure to a function?
19. Define binary files.
20. List some library functions for files
21. Define data structure.
22. Differentiate linear and non-liner data structures.
23. Define Stack.
24. What are the different operations can be performed on Stacks?
25. List some applications of stack.
26. What is the different format of arithmetic applications?
27. Define recursion.
28. Explain Towers of Hanoi.
29. Define Queue.
30. How do you represent a Queue in C.
31. Applications of Queue.
32. Different types of Queue.

33. Define Circular Queue.
34. How do you implement priority queue
35. List the operations of Queue.
36. Define linked list
37. How do you differentiate inked list from array
38. What is meant by dynamic variable?
39. Differentiate static and dynamic memory allocation.
40. List the operations of linked list.
41. How do you implement stack using linked list.
42. How do you implement queue using linked list.
43. What are the different types of linked lists?
44. Applications of inked lists.
45. Operation of doubly linked list.
46. Define binary tree.
47. How do you represent the binary tree.
48. How do you construct the binary search tree?
49. List the different traverses of a tree.
50. Applications of binary tree.

OBJECT ORIENTED PROGRAMMING WITH C++

10CS36

Object Oriented Programming with C++

(Common to CSE & ISE)

Sub. Code: 10CS36 IA Marks: 25
Hrs/Week: 04 Exam Hours: 03
Total Hours: 52 Exam Marks: 100

Unit I

1. Introduction to C++: A Review of Structures, Procedure-Oriented Programming Systems, Object-Oriented Programming Systems, Comparison of C++ with C, Console Input/Output in C++, Variables in C++, Reference Variables in C++, Function Prototyping, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions 4 Hours
2. Class and Objects: Introduction to Classes and Objects 2 Hours

Unit II

3. Class and Objects *contd*.: Member Functions and Member Data, Objects and Functions, Objects and Arrays, Namespaces, Nested Classes 6 Hours

Unit III

4. Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation, The set_new_handler() function
5. Constructors and Destructors: Constructors, Destructors, The Philosophy of OOPS 7 Hours

Unit IV

6. Inheritance: Introduction to Inheritance, Base Class and Deriv ed Class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access Specifiers, Different Kinds of Inheritance, Order of Invocation of Constructors and Destructors 6 Hours

Unit V

7. Virtual Functions and Dynamic Polymorphism: The Need for Virtual Functions, Virtual Functions, The Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors
8. Stream Handling: Streams, The Class Hierarchy of Handling Streams, Text and Binary Input/Output, Text Versus Binary Files, Text Input/Output, Binary Input/Output 6 Hours

Unit VI

9. Stream Handling *contd*.: Opening and Closing Files, Files as Objects of the fstream Class, File Pointer, Random Access to Files, Object Input/Output through Member Functions, Error Handling, Manipulators
10. 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 7 Hours

Unit VII

11. Operator Overloading contd.: Overloading the Relational Operators, Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators, Overloading the new and the delete Operators, Overloading the Subscript Operator, Overloading the Pointer-to-member (->) Operator (Smart Pointer) 7 Hours

Unit VIII

12. Type Conversion, New Style Casts, and RTTI
13. Templates: Introduction, Function Templates, Class Templates, The Standard Template Library (STL)
14. Exception Handling: Introduction, C-Style Handling of Error-generating Codes, C++ Style Solution – the try/throw/catch Construct, Limitation of Exception Handling 7 Hours

Text Book

1. Object-Oriented Programming with C++, Sourav Sahay, Oxford University Press, 2006 (Chapters 1 to 10)

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

OBJECT ORIENTED PROGRAMMING With C++
LESSON PLAN

SUBJECT CODE: 10CS36 **IA MARKS: 25**

HOURS/WEEK: 04 EXAM HOURS: -3

TOTAL HOURS: 052 EXAM MARKS: -100

SL NO	UNITS	HOUR NO	TOPICS TO BE COVERED
1	UNIT-1	1	**Part –A** Overview of C++, Sample C++ program
		2	Different Data Types, Operators, Expressions, and Statements,
		3	Arrays and Strings
		4	Pointers & User-Defined
		5	Types Function Components, argument passing
		6	Inline Functions, Function Overloading, Recursive Functions
2	Unit-2	7	Introduction to Classes and Objects with Examples. Class Specification,
		8	Scope resolution operator,
		9	Access members, Defining member functions, Data hiding,
		10	Constructors, Destructors, Parameterized constructors
		11	Static data members, Functions
		12	Internal Questions
3	Unit-3	13	Friend functions, Passing objects as arguments, Returning objects
		14	Arrays of objects, Dynamic objects
		15	Pointers to objects, Copy constructors
		16	Generic functions and classes, Applications
		17	Operator Overloading, Overloading Various Operators, Over Loading the Assignment Operators
		18	Operator overloading using friend functions such as +, -

		19	Pre-increment, Post-increment, [ ] etc., overloading <<, >>.
4	Unit-4	20	Introduction to Inheritance
		21	Base Class, Inheritance and protected members,
		22	Function Overriding
		23	Protected base class inheritance
		24	The Protected access Specifiers.
		25	Deriving by Different Access Specifiers.
		26	Different Kinds of Inheritance, Inheriting multiple base classes.
5	Unit-V	27	**Part-B** **Constructors**
		28	Constructors, Destructors and Inheritance
		29	Problems on Constructors , Destructors and Inheritance
		30	Passing parameters to base class constructors,
		31	Granting access
		32	Virtual base classes
6	Unit-6	33	Virtual functions, Polymorphism.
		34	Virtual function, Calling a Virtual function through a base class reference
		35	Virtual attribute is inherited
		36	Virtual functions are hierarchical,
		37	Pure virtual functions
		38	Abstract classes, Using virtual functions
		39	Early and late binding
7	Unit-7	40	**I/O System Basics, File I/0**
		41	C++ stream classes
		42	Formatted I/O,
		43	I/O manipulators,
		44	Fstream and the File classes
		45	File operations
8	Unit-8	46	**Exception Handling**
		47	The Standard Template Library (STL)
		48	Exception handling fundamentals
		49	Exception handling options

		50	STL: An overview
		51	Containers, Vectors, Lists, Maps.
		52	Revision of Previous Year Question Paper

Question Bank

Object-Oriented Programming with C++

1. Give an overview of structures in C.
2. Differentiate between procedure oriented programming and object oriented programming.
3. Compare C and C++
4. Give a few example of console input and output.
5. What is meant by reference variable in C++
6. What is function prototyping in C++. Give examples.
7. What is function overloading? Explain with example programmes.
8. Explain the concepts of default arguments in C++
9. Explain inline function with example code.
10. Explain the concepts of classes and objects in C++
11. What are the three access specifiers in C++? What is their functionality?
12. Explain the 'this' pointer
13. Data abstraction is a concept in cardinal OOPs. Explain
14. Explain the error operator.
15. What is meant by member functions and member data?
16. What is an inline member function?
17. Explain constant member functions.
18. What is mutable data member?
19. What is friend function in C++?
20. What is friend class-Explain?
21. Explain the concept of
22. What is meant by static member?
23. What is static member data and static member function?
24. What is an array of objects?
25. Explain the concept of namespaces.
26. What is nested classes? Explain.
27. Explain why friend functions do not contradict the principals of OOPs.
28. How do namespaces help in preventing pollution of the global namespace?
29. What is static memory allocation?
30. What is dynamic memory allocation? Explain. How it is different from static memory allocation?
31. Under what conditions does the use of dynamic memory allocation become Mendatory?

32. What is the syntax of the new operator for allocating memory for a single variable And an array of variables.
33. What is syntax of the delete operator for deallocating memory for a single Variable and for an array?
34. What is memory leak explain the concept?
35. What is the new-handler? What is meant by the Set-new-handler function?
36. What are constructors and distructors?
37. Why should the formal argument of a Copy Constructor be a reference object?
38. What is Copy Constructor Explain.
39. What is inheritance?
40. How does inheritance Compare with Containership?
41. What is multiplex? Multilevel inheritance?
42. What is meant by hybrid inheritance?
43. What is Virtual functions and polymorphism?
44. What is pure Virtual functions
45. What is Virtual destructors and Virtual constructors
46. What is streams Draw the class hierarchy for handling streams
47. What is manipulators? Explain with examples
48. What is meant by operator overloading? Explain .
49. What is meant by RTTI?
50. Explain function templates and class templates?
51. Explain the concept of exception handling?

DATA STRUCTURES WITH C/C++ LABORATORY

10CSL37

DATA STRUCTURES WITH C/C++ LABORATORY
(Common to CSE & ISE)

Subject Code: 10CSL37 **Exam Marks: 50**
Hours/Week : 03 **Total Hours : 42**
I.A. Marks : 25 **Exam Hours: 03**

1. Using circular representation for a polynomial, design, develop, and execute a program in C to accept two polynomials, add them, and then print the resulting polynomial.

2. Design, develop, and execute a program in C to convert a given valid parenthesized infix arithmetic expression to postfix expression and then to print both the expressions. The expression consists of 12 single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

3. Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).

4. Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations:

a. Insert b. Delete c. Display

5. Design, develop, and execute a program in C++ based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_Name (a string of characters), Basic_Salary (an integer) , All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic; Income Tax (IT) = 30% of the gross salary (= basic_Salary _ All_Allowance); Net_Salary = Basic_Salary + All_Allowances – IT)

6. Design, develop, and execute a program in C++ to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

i. STRING s1 = "VTU"

ii. STRING s2 = "BELGAUM"

iii. STIRNG s3 = s1 + s2; (Use copy constructor)

7. Design, develop, and execute a program in C++ to create a class called STACK using an array of integers and to implement the following operations by overloading the operators + and - :

i. s1=s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on to top of the stack.

ii. s1=s1- ; where s1 is an object of the class STACK and – operator pops off the top element.

iii Handle the STACK Empty and STACK Full conditions. Also display the contents of the stack after each operation, by overloading the operator <<.

8. Design, develop, and execute a program in C++ to create a class called LIST (linked list) with member functions to insert an element at the front of the list as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.

9. Design, develop, and execute a program in C to read a sparse matrix of integer values and to search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.

10. Design, develop, and execute a program in C to create a max heapof integers by accepting one element at a time and by inserting it immediately in to the heap. Use the array representation for the heap. Display the array at the end of insertion phase.

11. Design, develop, and execute a program in C to implement a doubly linked list where each node consists of integers. The program should support the following operations:

i. Create a doubly linked list by adding each node at the front.

ii. Insert a new node to the left of the node whose key value is read as an input.

iii. Delete the node of a given data if it is found, otherwise display appropriate message.

iv. Display the contents of the list.

(Note: Only either (a,b and d) or (a, c and d) may be asked in the examination)

12. Design, develop, and execute a program in C++ to create a class called DATE with methods to accept two valid dates in the form dd/mm/yy and to implement the following operations by overloading the operators + and -. After every operation the results are to be displayed by overloading the operator <<.

i. no_of_days = d1 – d2; where d1 and d2 are DATE objects, d1 >=d2 and no_of_days is an integer.

ii. d2 = d1 + no_of_days; where d1 is a DATE object and no_of_days is an integer.

13. Design, develop, and execute a program in C++ to create a class called OCTAL, which has the characteristics of an octal number. 14 Implement the following operations by writing an appropriate constructor and an overloaded operator +.

i. OCTAL h = x ; where x is an integer

ii. int y = h + k ; where h is an OCTAL object and k is an integer. Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

14. Design, develop, and execute a program in C++ to create a class called BIN_TREE that represents a Binary Tree, with member functions to perform inorder, preorder and postorder traversals. Create a BIN_TREE object and demonstrate the traversals.

Note: In the examination each student picks one question from a lot of all the 14 questions.

M.V. J. College of Engineering

Department of Computer Science & Engineering

Bangalore-560067

DATA STRUCTURES WITH C/C++ LABORATORY

Lesson Plan

Subject Code: 10CSL37 **I.A. Marks : 25**

Hours/Week : 03 Exam Hours: 03

Total Hours : 42 **Exam Marks: 50**

S.No	*Hours*	Topics to be covered
1.	3	Using circular representation for a polynomial, design, develop, and execute a program in C to accept two polynomials, add them, and then print the resulting polynomial.
2.	3	Design, develop, and execute a program in C to convert a given valid parenthesized infix arithmetic expression to postfix expression and then to print both the expressions. The expression consists of 12 single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).
3.	3	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).
4.	3	Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
5.	3	Design, develop, and execute a program in C++ based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_Name (a string of characters), Basic_Salary (an integer) , All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic; Income Tax (IT) = 30% of the gross salary (= basic_Salary _ All_Allowance); Net_Salary = Basic_Salary + All_Allowances – IT)

6.	3	Design, develop, and execute a program in C++ to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<. i. STRING s1 = "VTU" ii. STRING s2 = "BELGAUM" iii. STIRNG s3 = s1 + s2; (Use copy constructor)
7.	3	Design, develop, and execute a program in C++ to create a class called STACK using an array of integers and to implement the following operations by overloading the operators + and - : i. s1=s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on to top of the stack. ii. s1=s1- ; where s1 is an object of the class STACK and – operator pops off the top element. iii Handle the STACK Empty and STACK Full conditions. Also display the contents of the stack after each operation, by overloading the operator <<.
8.	3	Design, develop, and execute a program in C++ to create a class called LIST (linked list) with member functions to insert an element at the front of the list as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.
9.	3	Design, develop, and execute a program in C to read a sparse matrix of integer values and to search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.
10.	3	Design, develop, and execute a program in C to create a max heapof integers by accepting one element at a time and by inserting it immediately in to the heap. Use the array representation for the heap. Display the array at the end of insertion phase.
11.	3	Design, develop, and execute a program in C to implement a doubly linked list where each node consists of integers. The program should support the following operations: i. Create a doubly linked list by adding each node at the front. ii. Insert a new node to the left of the node whose key value is read as an input. iii. Delete the node of a given data if it is found, otherwise display appropriate message. iv. Display the contents of the list. (Note: Only either (a,b and d) or (a, c and d) may be asked in the examination)

12.	3	Design, develop, and execute a program in C++ to create a class called DATE with methods to accept two valid dates in the form dd/mm/yy and to implement the following operations by overloading the operators + and -. After every operation the results are to be displayed by overloading the operator <<. i. no_of_days = d1 – d2; where d1 and d2 are DATE objects, d1 >=d2 and no_of_days is an integer. ii. d2 = d1 + no_of_days; where d1 is a DATE object and no_of_days is an integer.
13.	3	Design, develop, and execute a program in C++ to create a class called OCTAL, which has the characteristics of an octal number. 14 Implement the following operations by writing an appropriate constructor and an overloaded operator +. i. OCTAL h = x ; where x is an integer ii. int y = h + k ; where h is an OCTAL object and k is an integer. Display the OCTAL result by overloading the operator <<. Also display the values of h and y.
14.	3	Design, develop, and execute a program in C++ to create a class called BIN_TREE that represents a Binary Tree, with member functions to perform inorder, preorder and postorder traversals. Create a BIN_TREE object and demonstrate the traversals.

Signature of HOD **Signature of Staff**

ELECTRONIC CIRCUITS & LOGIC DESIGN LABORATORY

10CSL38

ELECTRONIC CIRCUITS & LOGIC DESIGN LABORATORY
SYLLABUS

Sub Code : 06CSL38 **IA Marks : 25**
Hrs / Week : 03 Exam Hours : 03
Total Hrs : 42 **Exam Marks : 50**

PART – A

1. a. To study the working of positive clipper, double-ended clipper and positive clamper using diodes.
b. To build and simulate the above circuits using a simulation package
2. a. To determine the frequency response, input impedance, output impedance, and bandwidth of a CE amplifier.
b. To build the CE amplifier circuit using a simulation package and determine the voltage gain for two different values of supply voltage and for two different values of emitter resistance.
3. a. To determine the drain characteristics and transconductance characteristics of an enhancement-mode MOSFET.
b. To implement a CMOS inverter using a simulation package and verify its truthtable.
4. a. To design and implement a Schmitt trigger using Op-Amp for given UTP and LTP values.
b. To implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values.
5. a. To design and implement a rectangular waveform generator (Op- Amp relaxation oscillator) for given frequency.
b. To implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and observe the change in frequency when all resistor values are doubled.
6. To design and implement an astable multivibrator circuit using 555 timer for a given frequency and duty cycle.
7. To implement a +5V regulated power supply using full-wave rectifier and 7805 IC regulator in simulation package. Find the output ripple for different values of load current.

PART – B

1. a. Given any 4-variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
b. Write the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.
2. a. Realize a full adder using 3-to-8 decoder IC and 4 input NAND gates.
b. Write the Verilog/VHDL code for a full adder. Simulate and verify its working.
3. a. Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table.
b. Write the Verilog/VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
4. a. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs.
b. Write the Verilog/VHDL code for mod-8 up counter. Simulate and verify its working.
5. a. Design and implement a ring counter using 4-bit shift register.
b. Write the Verilog/VHDL code for switched tail counter. Simulate and verify its working.
6. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9).
7. Design a 4-bit R-2R ladder D/A converter using Op-Amp. Determine its accuracy and resolution.

Note 1. *Any simulation package like MultiSim/Pspice etc. may be used.*

M.V. J. College of Engineering
Department of Computer Science & Engineering
Bangalore-560067

ELECTRONIC CIRCUITS & LOGIC DESIGN LABORTORY
Lesson Plan

Subject Code: 06CSL38 **I.A. Marks: 25**
Hours/Week: 03 Exam Hours: 03
Total Hours: 42 **Exam Marks: 50**

S.No	Chapter	Hour	Topics to be covered
1.	Positive Clipper, Double Ended Clipper and Positive Clamper	3	a) To Study The Working of Positive Clipper, Double Ended Clipper and Positive Clamper using Diodes. b) To Build and Simulate the above Circuits using Simulation Package
2.	CE amplifier	3	a) To determine the frequency response, input impedance, output impedance, and bandwidth of a CE amplifier. b) To Build the CE amplifier circuit using a simulation package and determine the voltage gain for two different values of supply voltage and for two different values of emitter resistance.
3.	**E-MOSFET characteristics**	3	a) To determine the drain characteristics and transconductance characteristics of an enhancement-mode MOSFET. b) To implement a CMOS inverter using a simulation package and verify its truth table.
4.	**Schmitt trigger using OP-AMP.**	3	a) To design and implement a Schmitt trigger using OP-AMP for given UTP and LTP values. b) To implement a Schmitt trigger using OP-AMP using a simulation package for two sets of UTP and LTP values.

5.	**Rectangular waveform generator**	3	a) To design and implement a rectangular waveform generator (OP-AMP relaxation oscillator) for given frequency. b) To implement a rectangular waveform generator (OP-AMP relaxation oscillator) using a simulation package and observe the change in frequency when all resistor values are doubled.
6.	**Astable multivibrator using 555 timer.**	3	To design and implement a suitable multivibrator circuit using 555 timers for a given frequency and duty cycle.
7.	**7805 IC regulator**	3	To implement a +5v regulated power supply using full-wave rectifier and 7805 IC regulators in simulation package. Find the output ripple for different values of load current.
8.	**Multiplexer IC**	3	**a) Given any 4-varible logic expression, simply using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.** b) Write the Verilog / VHDL code for an 8:1 multiplexer, Simulate and verify its working.
9.	**Full adder using 3-to 8 decoder IC**	3	a) Realize a full adder using 3-to-8 decoder IC and 4 inputs NAND gate. b) Write the Verilog/VHDL code for a full adder. Simulate and verify its working.
10.	**J-K master/slave flip flop**	3	**a) Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table.** b) Write the Verilog/VHDL code for a full adder. Simulate and verify its working.
11.	**Mod-n synchronous up counter**	3	**a) Design and implement a mod-m (n<8)synchronous up counter using J-K filp flop ICs.** b) Write the Verilog/VHDL code for mod-8 up counter. Simulate and verify its working.

12.	**Ring counter**	3	a) Design and implement a ring counter using 4-bit shift register. b) Write the Verilog/VHDL code for switched tail counter. Simulate and verify its working.
13.	**Asynchronous counter**	3	Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9).
14.	**R-2R ladder D/A Converter**	3	Design a 4-bit R-2R ladder D/A converter using OP-Amp. Determine its accuracy and resolution.
15	**Part A-Test-HW**	3	
16	**Part A-Test-SW**	2	
17	**Part B-Test-HW**	3	
18	**Part B-Test-SW**	2	