Analog and Digital Electronics

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS32**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Recall and Recognize construction and characteristics of JFETs and MOSFETs and differentiate with BJT
- Demonstrate and Analyze Operational Amplifier circuits and their applications
- Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
- Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops.
- Describe, Design and Analyze Synchronous and Asynchronous Sequential
- Explain and design registers and Counters, A/D and D/A converters.

Module -1	**Teaching Hours**
Field Effect Transistors: Junction Field Effect Transistors, MOSFETs, Differences between JFETs and MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices. Wave-Shaping Circuits: Integrated Circuit(IC) Multivibrators. **Introduction to Operational Amplifier**: Ideal v/s practical Opamp, Performance Parameters, **Operational Amplifier Application Circuits**:Peak Detector Circuit, Comparator, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter. **Text book 1:- Ch 5: 5.2, 5.3, 5.5, 5.8, 5.9, 5.1.Ch13: 13.10.Ch 16: 16.3, 16.4. Ch 17: 7.12, 17.14, 17.15, 17.18, 17.19, 17.20, 17.21.**	**10 Hours**
Module -2	
The Basic Gates: Review of Basic Logic gates, Positive and Negative Logic, Introduction to HDL. **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. **Text book 2:- Ch 2: 2.4, 2.5. Ch3: 3.2 to 3.11.**	**10 Hours**
Module – 3	

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit **Flip- Flops:** RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs. **Text book 2:- Ch 4:- 4.1 to 4.9, 4.11, 4.12, 4.14.Ch 6:-6.7, 6.10.Ch 8:- 8.1 to 8.5.**	**10 Hours**
Module-4	
Flip- Flops: FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. **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. **Counters:** Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. (**Text book 2:- Ch 8: 8.6, 8.8, 8.9, 8.10, 8.13. Ch 9: 9.1 to 9.8. Ch 10: 10.1 to 10.4**	**10 Hours**
Module-5	
Counters: Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. **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 2:- Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10.**	**10 Hours**
Course outcomes:	
After Studying this course, students will be able to • Acquire knowledge of o JFETs and MOSFETs , Operational Amplifier circuits and their applications. o Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique. o Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors. o Working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters. • Analyze the performance of o JFETs and MOSFETs , Operational Amplifier circuits o Simplification Techniques using Karnaugh Maps, Quine McClusky Technique. o Synchronous and Asynchronous Sequential Circuits. Apply the knowledge gained in the design of Counters, Registers and A/D & D/A converters	
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Design/Development of Solutions(partly) 3. Modern Tool Usage 4. Problem Analysis	

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015

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. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson, 2008.

DATA STRUCTURES AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS33**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Explain fundamentals of data structures and their applications essential for programming/problem solving
- Analyze Linear Data Structures: Stack, Queues, Lists
- Analyze Non-Linear Data Structures: Trees, Graphs
- Analyze and Evaluate the sorting & searching algorithms
- Assess appropriate data structure during program development/Problem Solving

Module -1	**Teaching Hours**
Introduction: Data Structures, Classifications (Primitive & Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays, **Array Operations**: Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices. **Strings:** Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples. **Text 1: Ch 1: 1.2, Ch 2: 2.2 -2.7** **Text 2: Ch 1: 1.1 -1.4, Ch 3: 3.1-3.3,3.5,3.7, Ch 4: 4.1-4.9,4.14** **Ref 3: Ch 1: 1.4**	**10 Hours**
Module -2	
Stacks and Queues ***Stacks:*** Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples. **Text 1: Ch 3: 3.1 -3.7** **Text 2: Ch 6: 6.1 -6.3, 6.5, 6.7-6.10, 6.12, 6.13**	**10 Hours**
Module - 3	

Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples **Text 1: Ch 4: 4.1 -4.8 except 4.6** **Text 2: Ch 5: 5.1 – 5.10**	**10 Hours**
Module-4	
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples **Text 1: Ch 5: 5.1 –5.5, 5.7** **Text 2: Ch 7: 7.1 – 7.9**	**10 Hours**
Module-5	
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. **Sorting and Searching**: Insertion Sort, Radix sort, Address Calculation Sort. **Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Files and Their Organization:** Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing **Text 1: Ch 6: 6.1 –6.2, Ch 7:7.2, Ch 8:8.1-8.3** **Text 2: Ch 8: 8.1 – 8.7, Ch 9:9.1-9.3,9.7,9.9** **Reference 2: Ch 16: 16.1 - 16.7**	**10 Hours**
Course outcomes:	
After studying this course, students will be able to: • Acquire knowledge of - Various types of data structures, operations and algorithms. - Sorting and searching operations. - File structures. • Analyse the performance of - Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques. • Implement all the applications of Data structures in a high-level language. • Design and apply appropriate data structures for solving computing problems.	
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Design/Development of Solutions 3. Conduct Investigations of Complex Problems 4. Problem Analysis	

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Fundamentals of Data Structures in C - Ellis Horowitz and Sartaj Sahni, 2nd edition, Universities Press,2014
2. Data Structures - Seymour Lipschutz, Schaum's Outlines, Revised 1st edition, McGraw Hill, 2014

Reference Books:

1. Data Structures: A Pseudo-code approach with C –Gilberg & Forouzan, 2nd edition, Cengage Learning, 2014.
2. Data Structures using C, , Reema Thareja, 3rd edition Oxford press, 2012.
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, McGraw Hill, 2013.
4. Data Structures using C - A M Tenenbaum, PHI, 1989.
5. Data Structures and Program Design in C - Robert Kruse, 2nd edition, PHI, 1996.

COMPUTER ORGANIZATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS34**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives:

This course will enable students to

- Understand the basics of computer organization: structure and operation of computers and their peripherals.
- Understand the concepts of programs as sequences or machine instructions.
- Expose different ways of communicating with I/O devices and standard I/O interfaces.
- Describe hierarchical memory systems including cache memories and virtual memory.
- Describe arithmetic and logical operations with integer and floating-point operands.
- Understand basic processing unit and organization of simple processor, concept of pipelining and other large computing systems.

Module -1	**Teaching Hours**
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. **Machine Instructions and Programs:** Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions **Textbook 1: Ch 1: 1.3, 1.4, 1.6.1, 1.6.2, 1.6.4, 1.6.7. Ch 2: 2.2 to 2.10, 2.12**	**10Hours**
Module -2	
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB. **Textbook 1: Ch 4: 4.1, 4.2: 4.2.1 to 4.2.5, 4.4 to 4.7.**	**10 Hours**
Module – 3	
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories, Secondary Storage. **Textbook 1: Ch 5: 5.1 to 5.4, 5.5.1, 5.5.2, 5.6, 5.7, 5.9**	**10 Hours**
Module-4	

Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations. **Textbook 1: Ch 2: 2.1, Ch 6: 6.1 to 6.7**	**10 Hours**
Module-5	
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. **Embedded Systems and Large Computer Systems:** Examples of Embedded Systems, Processor chips for embedded applications, Simple Microcontroller. **The structure of General-Purpose Multiprocessors**. **Textbook 1: Ch 7: 7.1 to 7.5, Ch 9:9.1 to 9.3, Ch 12:12.3**	**10 Hours**

Course outcomes:

After studying this course, students will be able to:

- Acquire knowledge of
  - The basic structure of computers & machine instructions and programs, Addressing Modes, Assembly Language, Stacks, Queues and Subroutines.
  - Input/output Organization such as accessing I/O Devices, Interrupts.
  - Memory system basic Concepts, Semiconductor RAM Memories, Static memories, Asynchronous DRAMS, Read Only Memories, Cache Memories and Virtual Memories.
  - Some Fundamental Concepts of Basic Processing Unit, Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control and Micro programmed Control.
  - Pipelining, embedded and large computing system architecture.
- Analyse and design arithmetic and logical units.
- Apply the knowledge gained in the design of Computer.
- Design and evaluate performance of memory systems
- Understand the importance of life-long learning

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Life-Long Learning

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition, Tata McGraw Hill, 2002.

Reference Books:

1. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.

UNIX AND SHELL PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER – III

Subject Code	**15CS35**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Understand the UNIX Architecture, File systems and use of basic Commands.
- Use of editors and Networking commands.
- Understand Shell Programming and to write shell scripts.
- Understand and analyze UNIX System calls, Process Creation, Control & Relationship.

Module -1	**Teaching Hours**
Introduction, Brief history. Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. The login prompt. General features of Unix commands/ command structure. Command arguments and options. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The man command knowing more about Unix commands and using Unix online manual pages. The man with keyword option and whatis. The more command and using it with other commands. Knowing the user terminal, displaying its characteristics and setting characteristics. Managing the non-uniform behaviour of terminals and keyboards. The root login. Becoming the super user: su command. The /etc/passwd and /etc/shadow files. Commands to add, modify and delete users. **Topics from chapter 2 , 3 and 15 of text book 1,chapter 1 from text book 2**	**10Hours**
Module -2	
Unix files. Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. File attributes and permissions and knowing them. The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. **Topics from chapters 4, 5 and 6 of text book 1**	**10Hours**

Module – 3	
The vi editor. Basics. The .exrc file. Different ways of invoking and quitting vi. Different modes of vi. Input mode commands. Command mode commands. The ex mode commands. Illustrative examples Navigation commands. Repeat command. Pattern searching. The search and replace command. The set, map and abbr commands. Simple examples using these commands. The shells interpretive cycle. Wild cards and file name generation. Removing the special meanings of wild cards. Three standard files and redirection. Connecting commands: Pipe. Splitting the output: tee. Command substitution. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions. **Topics from chapters 7, 8 and 13 of text book 1. Topics from chapter 2 and 9 ,10 of text book 2**	**10Hours**
Module-4	
Shell programming. Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. File inodes and the inode structure. File links – hard and soft links. Filters. Head and tail commands. Cut and paste commands. The sort command and its usage with different options. The umask and default file permissions. Two special files /dev/null and /dev/tty. **Topics from chapter 11, 12, 14 of text book 1,chapter 17 from text book2**	**10Hours**
Module-5	
Meaning of a process. Mechanism of process creation. Parent and child process. The ps command with its options. Executing a command at a specified point of time: at command. Executing a command periodically: cron command and the crontab file.. Signals. The nice and nohup commands. Background processes. The bg and fg command. The kill command. The find command with illustrative example. Structure of a perl script. Running a perl script. Variables and operators. String handling functions. Default variables - $_ and $. – representing the current line and current line number. The range operator. Chop() and chomp() functions. Lists and arrays. The @-variable. The splice operator, push(), pop(), split() and join(). File handles and handling file – using open(), close() and die () functions.. Associative arrays – keys and value functions. Overview of decision making loop control structures – the foreach. Regular expressions – simple and multiple search patterns. The match and substitute operators. Defining and using subroutines. **Topics from chapter 9 and 19 of text book 1. Topics from chapter 11 of reference book 1**	**10Hours**

Course outcomes:

After studying this course, students will be able to:

- Explain multi user OS UNIX and its basic features
- Interpret UNIX Commands, Shell basics, and shell environments
- Design and develop shell programming, communication, System calls and terminology.
- Design and develop UNIX File I/O and UNIX Processes.
- Perl script writing

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Environment and Sustainability
3. Design/Development of Solutions

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Sumitabha Das., Unix Concepts and Applications., 4th Edition., Tata McGraw Hill
2. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009.

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming**,** Pearson Education.
2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2nd Edition , Wiley,2014.

DISCRETE MATHEMATICAL STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER – III

Subject Code	**15CS36**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS – 04

Course objectives: This course will enable students to

- Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science.
- Understand and apply logic, relations, functions, basic set theory, countability and counting arguments, proof techniques,
- Understand and apply mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory
- Understand and apply graph theory and mathematical proof techniques.

Module -1	**Teaching Hours**
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems, **Textbook 1: Ch 2**	**10Hours**
Module -2	
Properties of the Integers: Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions. **Fundamental Principles of Counting:** The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition, **Textbook 1: Ch 4: 4.1, 4.2 Ch 1.**	**10 Hours**
Module – 3	
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. **Properties of Relations**, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **Textbook 1: Ch 5:5.1 to 5.3, 5.5, 5.6, Ch 7:7.1 to 7.4**	**10 Hours**
Module-4	

The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials. **Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients. **Textbook 1: Ch 8: 8.1 to 8.4, Ch 10:10.1 to 10.2**	**10 Hours**
Module-5	
Introduction to Graph Theory: Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits , **Trees**: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes **Textbook 1: Ch 11: 11.1 to 11.3, Ch 12: 12.1 to 12.4**	**10 Hours**

Course outcomes:

After studying this course, students will be able to:

1. Verify the correctness of an argument using propositional and predicate logic and truth tables.
2. Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability.
3. Solve problems involving recurrence relations and generating functions.
4. Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction.
5. Explain and differentiate graphs and trees

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Conduct Investigations of Complex Problems

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, , 5th Edition, Pearson Education. 2004.

Reference Books:

1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 2016
2. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.
3. Jayant Ganguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010.
4. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.
5. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

ANALOG AND DIGITAL ELECTRONICS LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL37**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 02

Course objectives: This laboratory course enable students to get practical experience in design, assembly and evaluation/testing of

- Analog components and circuits including Operational Amplifier, Timer, etc.
- Combinational logic circuits.
- Flip - Flops and their operations
- Counters and Registers using Flip-flops.
- Synchronous and Asynchronous Sequential Circuits.
- A/D and D/A Converters

Descriptions (if any)

Any simulation package like MultiSim / P-spice /Equivalent software may be used.
Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-upon Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These **TWO Laboratory sessions** are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.

Laboratory Experiments:

1. a) Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
   b) Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.

2. a) Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
   b) Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.

3. Design and implement an Astable multivibrator circuit using 555 timer for a given frequency and duty cycle.

NOTE: hardware and software results need to be compared

Continued:

4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.

5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
   b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.

6. a) Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.

7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.

8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
   b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.

9. a) Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
   b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.

10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447).

11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

Study experiment

12. To study 4-bitALU using IC-74181.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic circuits.
- Design and demonstrate various types of counters and Registers using Flip-flops
- Use simulation package to design circuits.
- Understand the working and implementation of ALU.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:
   a) For questions having part a only- Procedure + Conduction + Viva:**20 + 50 +10 =80 Marks**
   b) For questions having part a and b
   Part a- Procedure + Conduction + Viva:1**0 + 35 +05= 50 Marks**
   Part b- Procedure + Conduction + Viva:1**0 + 15 +05= 30 Marks**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

DATA STRUCTURES LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL38**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives:

This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Asymptotic performance of algorithms.
- Linear data structures and their applications such as Stacks, Queues and Lists
- Non-Linear Data Structures and their Applications such as Trees and Graphs
- Sorting and Searching Algorithms

Descriptions (if any)

Implement all the experiments in C Language under Linux / Windows environment.

Laboratory Experiments:

1. Design, Develop and Implement a menu driven Program in C for the following **Array** operations
    a. Creating an Array of **N** Integer Elements
    b. Display of Array Elements with Suitable Headings
    c. Inserting an Element (**ELEM**) at a given valid Position (**POS**)
    d. Deleting an Element at a given valid Position(**POS**)
    e. Exit.

    Support the program with functions for each of the above operations.

2. Design, Develop and Implement a Program in C for the following operationson **Strings**
    a. Read a main String (**STR),** a Pattern String (**PAT**) and a Replace String (**REP**)
    b. Perform Pattern Matching Operation: Find and Replace all occurrences of **PAT** in **STR** with **REP** if **PAT** exists in **STR.** Report suitable messages in case **PAT** does not exist in **STR**

    Support the program with functions for each of the above operations. Don't use Built-in functions.

3. Design, Develop and Implement a menu driven Program in C for the following operations on **STACK** of Integers (Array Implementation of Stack with maximum size **MAX**)
    a. ***Push*** an Element on to Stack
    b. ***Pop*** an Element from Stack
    c. Demonstrate how Stack can be used to check ***Palindrome***
    d. Demonstrate ***Overflow*** and ***Underflow*** situations on Stack

e. Display the status of Stack
f. Exit

Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: **+, -, *, /, %(Remainder), ^(Power)** and **alphanumeric** operands**.**

5. Design, Develop and Implement a Program in C for the following Stack Applications
   a. Evaluation of **Suffix expression** with single digit operands and operators: **+, -, *, /, %, ^**
   b. Solving **Tower of Hanoi** problem with **n** disks

6. Design, Develop and Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)
   a. Insert an Element on to Circular QUEUE
   b. Delete an Element from Circular QUEUE
   c. Demonstrate ***Overflow*** and ***Underflow*** situations on Circular QUEUE
   d. Display the status of Circular QUEUE
   e. Exit

   Support the program with appropriate functions for each of the above operations

Continued:

7. Design, Develop and Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: ***USN, Name, Branch, Sem, PhNo***
   a. Create a **SLL** of **N** Students Data by using ***front insertion***.
   b. Display the status of **SLL** and count the number of nodes in it
   c. Perform Insertion / Deletion at End of **SLL**
   d. Perform Insertion / Deletion at Front of **SLL(Demonstration of stack)**
   e. Exit

8. Design, Develop and Implement a menu driven Program in C for the following operations on **Doubly Linked List (DLL)** of Employee Data with the fields: ***SSN, Name, Dept, Designation, Sal, PhNo***
   a. Create a **DLL** of **N** Employees Data by using ***end insertion***.
   b. Display the status of **DLL** and count the number of nodes in it
   c. Perform Insertion and Deletion at End of **DLL**
   d. Perform Insertion and Deletion at Front of **DLL**
   e. Demonstrate how this **DLL** can be used as **Double Ended Queue**
   f. Exit

9. Design, Develop and Implement a Program in C for the following operationson **Singly Circular Linked List (SCLL)** with header nodes
    a. Represent and Evaluate a Polynomial **$P(x,y,z) = 6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$**
    b. Find the sum of two polynomials **POLY1(x,y,z)** and **POLY2(x,y,z)** and store the result in **POLYSUM(x,y,z)**

    Support the program with appropriate functions for each of the above operations
10. Design, Develop and Implement a menu driven Program in C for the following operations on **Binary Search Tree (BST)** of Integers
    a. Create a BST of **N** Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
    b. Traverse the BST in Inorder, Preorder and Post Order
    c. Search the BST for a given element (**KEY**) and report the appropriate message
    e. Exit

11. Design, Develop and Implement a Program in C for the following operations on **Graph(G)** of Cities
    a. Create a Graph of **N** cities using Adjacency Matrix.
    b. Print all the nodes **reachable** from a given starting node in a digraph using DFS/BFS method

12. Given a File of **N** employee records with a set **K** of Keys(4-digit) which uniquely determine the records in file **F**. Assume that file **F** is maintained in memory by a Hash Table(HT) of **m** memory locations with **L** as the set of memory addresses (2-digit) of locations in HT. Let the keys in **K** and addresses in **L** are Integers. Design and develop a Program in C that uses Hash function **H: K $\rightarrow$L** as H(**K**)=**K** mod **m (remainder** method), and implement hashing technique to map a given key **K** to the address space **L.** Resolve the collision (if any) using **linear probing**.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Analyze and Compare various linear and non-linear data structures
- Code, debug and demonstrate the working nature of different types of data structures and their applications
- Implement, analyze and evaluate the searching and sorting algorithms
- Choose the appropriate data structure for solving real world problems

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (**TWELVE** nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva:**20 + 50 +10 (80)**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**