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

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.


Module -2


Text book 2:- Ch 2: 2.4, 2.5. Ch3: 3.2 to 3.11.

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.


<table>
<thead>
<tr>
<th>Module-4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flip-Flops:</strong> 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. <strong>Counters:</strong> Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus.</td>
</tr>
<tr>
<td>(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)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Text book 2:- Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10.)</td>
</tr>
</tbody>
</table>

**Course outcomes:**
- Acquire knowledge of
  - JFETs and MOSFETs, Operational Amplifier circuits and their applications.
  - Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique,
  - Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors.

  - Analyze the performance of
    - JFETs and MOSFETs, Operational Amplifier circuits
    - Simplification Techniques using Karnaugh Maps, Quine McClusky Technique.
    - 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:**


**Reference Books:**

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

Introduction:

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

Module -2

Stacks and Queues

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

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

<table>
<thead>
<tr>
<th>10 Hours</th>
</tr>
</thead>
</table>

#### 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

<table>
<thead>
<tr>
<th>10 Hours</th>
</tr>
</thead>
</table>

#### 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

<table>
<thead>
<tr>
<th>10 Hours</th>
</tr>
</thead>
</table>

**Course outcomes:**

- 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:**


**Reference Books:**

### COMPUTER ORGANIZATION

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

#### SEMESTER - III

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>15CS34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>Exam Marks</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>Exam Hours</td>
</tr>
</tbody>
</table>

**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

**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

### 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.

### 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

### Module-4
### Arithmetic


**Textbook 1:** Ch 2: 2.1, Ch 6: 6.1 to 6.7

**Module-5**

**Basic Processing Unit:** Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Microprogrammed Control.


**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 Microprogrammed 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


### Reference Books

# UNIX AND SHELL PROGRAMMING

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

<table>
<thead>
<tr>
<th>SEMESTER – III</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>15CS35</td>
<td>20</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>04</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Exam Hours</td>
<td>03</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10Hours</td>
</tr>
</tbody>
</table>

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**

## Module -2

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10Hours</td>
</tr>
</tbody>
</table>


**Topics from chapters 4, 5 and 6 of text book 1**
<table>
<thead>
<tr>
<th>Module – 3</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Module-4</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Module-5</th>
</tr>
</thead>
</table>
### 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:


### Reference Books:

### 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.

<table>
<thead>
<tr>
<th>Module -1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fundamentals of Logic:</strong> 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, <strong>Textbook 1: Ch 2</strong></td>
<td>10 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module -2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties of the Integers:</strong> Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions. <strong>Fundamental Principles of Counting:</strong> The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition, <strong>Textbook 1: Ch 4: 4.1, 4.2 Ch 1.</strong></td>
<td>10 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module -3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relations and Functions:</strong> Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. <strong>Properties of Relations,</strong> Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. <strong>Textbook 1: Ch 5:5.1 to 5.3, 5.5, 5.6, Ch 7:7.1 to 7.4</strong></td>
<td>10 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module-4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>****</td>
<td>10 hours</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Textbook 1: Ch 8: 8.1 to 8.4, Ch 10:10.1 to 10.2</th>
<th>10 Hours</th>
</tr>
</thead>
</table>

### 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

<table>
<thead>
<tr>
<th>Textbook 1: Ch 11: 11.1 to 11.3, Ch 12: 12.1 to 12.4</th>
<th>10 Hours</th>
</tr>
</thead>
</table>

### 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:


### Reference Books:

### Laboratory Code: 15CSL37

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>IA Marks</th>
<th>IA Marks</th>
<th>Number of Lecture Hours</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>01L + 02P</td>
<td>20</td>
<td></td>
<td>40</td>
<td>80</td>
<td>03</td>
</tr>
</tbody>
</table>

**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-up on 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-up on 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:


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.


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-bit ALU 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 Train 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: 10 + 35 + 05 = 50 Marks
      Part b- Procedure + Conduction + Viva: 10 + 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

<table>
<thead>
<tr>
<th>Laboratory Code</th>
<th>15CSL38</th>
<th>IA Marks</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>01L + 02P</td>
<td>Exam Marks</td>
<td>80</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

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 operations on 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
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 operations on Singly Circular Linked List (SCLL) with header nodes
   a. Represent and Evaluate a Polynomial \( P(x, y, z) = 6x^2y^2z^4 + 3x^3yz + 2xy^5z - 2xyz^3 \)
   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.