

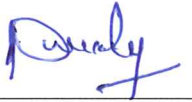








AMR/CS/BOS/2024-25/MOM/1

Date : 23-09-2024

## DEPARTMENT OF CYBER SECURITY

### MINUTES OF MEETING - BOARD OF STUDIES (BOS)

The Meeting of the Board of Studies of Cyber Security was held on 23<sup>rd</sup> September 2024 at 10.30 AM through online (Zoom Meeting Platform). The following members were attended the online meeting.

S.No	Name of the Member	Designation/occupation	Category	Signature
1	Mr. A Madhava Reddy	Head of the Department	Chairman	
2	Dr. A S N Chakravarthy	Professor, CSE Department, UCEK, JNTUK Kakinada	University Nominee	
3	Dr. E Vamsidhar	Professor & Associate Dean, Department of CSE , K L Univeristy	Subject experts outside parent university	
4	Dr. K Thirupathi Rao	Professor, Gitam Demeed to be University, Vishakapatnam		
5	Mr. M Rosi Reddy	IT- Director, Sanofi US	Industrialist	
6	Mrs. A Sai Pujitha	Asst. Professor	Faculty Members	
7	Mr . G Mohan Singh Yadav	Asst. Professor		
8	Mr. K Rama Linga Reddy	Asst. Professor		
9	Mr. S. Azeez	Software Engineer Techniq Creations Pvt. Ltd.	Alumni Member	

The Meeting began with chairman, Board of studies extending a warm welcome to all the members of participating in the meeting.

The following points were discussed and approved during the meeting

1. The following proposed AMR 24 Course Structure and the detailed syllabi of I-I, I-II were presented, discussed and approved.

## 1. For Group – A

GROUP –A – COURSES (CSE ,EEE)					
I Year – I SEM					
S.No.	Title	Credits	S. No.	Title	Credits
1	Communicative English	2	6	Communicative English Lab	1
2	Engineering Chemistry/Chemistry/Fundamental Chemistry	3	7	Engineering Chemistry/ Chemistry/Fundamental Chemistry Lab	1
3	Linear Algebra & Calculus	3	8	Engineering Workshop	1.5
4	Basic Civil & Mechanical Engineering	3	9	<b>Computer Programming Lab</b>	1.5
5	<b>Introduction to Programming</b>	3	10	Health and wellness, Yoga and Sports	0.5
<b>I SEM - TOTAL CREDITS</b>					<b>20.5</b>
I Year – II SEM					
1	Engineering Physics	3	6	<b>IT Workshop</b>	1
2	Differential Equations & Vector Calculus	3	7	Engineering Physics Lab	1
3	Basic Electrical and Electronics Engineering	3	8	Electrical and Electronics Engineering Workshop	1.5
4	Engineering Graphics	3	9	<b>Data Structures Lab</b> / Electrical Circuits Lab	1.5
5	<b>Data Structures</b> / Electrical Circuit Analysis – I (Branch specific)	3	10	NSS/NCC/Scouts & Guides/Community Service	0.5
<b>II SEM - TOTAL CREDIT</b>					<b>19.5</b>

**2. For Group –B**

<b>GROUP –B – COURSES (Cyber Security ,ECE,MECH,CE, AGE)</b>					
<b>I Year – I SEM</b>					
<b>S.No.</b>	<b>Title</b>	<b>Credits</b>	<b>S.No.</b>	<b>Title</b>	<b>Credits</b>
1	Engineering Physics	3	6	<b>IT Workshop</b>	1
2	Linear Algebra & Calculus	3	7	Engineering Physics Lab	1
3	Basic Electrical & Electronics Engineering	3	8	Electrical & Electronics Engineering Workshop	1.5
4	Engineering Graphics	3	9	<b>Computer Programming Lab</b>	1.5
5	<b>Introduction to Programming</b>	3	10	NSS/NCC/Scouts & Guides/Community Service	0.5
<b>I SEM - TOTAL CREDITS</b>					<b>20.5</b>
<b>I Year – II SEM</b>					
1	Communicative English	2	6	Communicative English Lab	1
2	Engineering Chemistry / Chemistry / Fundamental Chemistry	3	7	Engineering Chemistry / Chemistry /Fundamental Chemistry Lab	1
3	Differential Equations & Vector Calculus	3	8	Engineering Workshop	1.5
4	Basic Civil & Mechanical Engineering	3	9	Engineering Mechanics & Building Practices Lab Engineering Mechanics Lab / Network Analysis and Simulation Lab / <b>Data structures Lab</b>	1.5
5	Engineering Mechanics/ Network Analysis/ <b>Data structures</b> (Branch specific)	3	10	Health and wellness, Yoga and Sports	0.5
<b>II SEM - TOTAL CREDITS</b>					<b>19.5</b>

3. The syllabus for Introduction to Programming (**Annexure -1**), Computer Programming Lab (**Annexure -2**) , Data Structures (**Annexure -3** ), Data Structures Lab (**Annexure -4** ), IT Workshop(**Annexure -5**) and has been approved by the Board of Studies.

## The following points were suggested for future possible implementations:

### Prescribed Textbooks:

Advised to add recent prescribed textbooks or updated editions for the course.

### C Programming:

Suggested using "*Let Us C*" by "Yashwanth Kanetkar" for additional examples and better understanding of C programming.

### References:

More references are recommended to be added for various topics.

### Computer Programming Labs:

The university suggested conducting the Computer Programming labs on a Linux platform using **Putty**.

### IT Workshop:

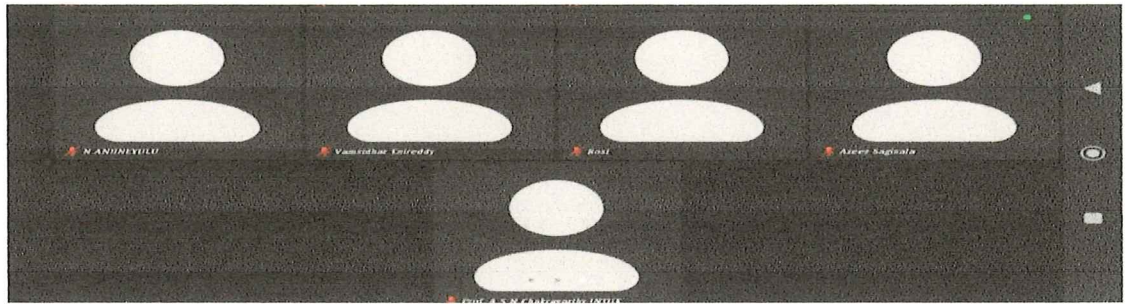
Recommended adding networking concepts such as LAN, shared drives, LAN connections, browser settings, and search engines likewise.


### Information Sharing:

All the updated information should be shared at least 15 days before the next Board of Studies (BOS) meeting.


The BOS chairman concluded the session and informed that the suggested points will be implemented and mail the same for approval and requested the experts to approve and ended with Vote of Thanks.








**A.M. REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY**  
Approved by AICTE, New Delhi, Affiliated to JNTU - Kakinada, Accredited by NAAC  
*An Autonomous Institution*  
Web : [www.amreddyengineering.ac.in](http://www.amreddyengineering.ac.in), E-mail: [principal.amreddyengineering@gmail.com](mailto:principal.amreddyengineering@gmail.com) Ph : 98664 14252  
Vinukonda Road, Petalurivaripalem, Narasaraopet, Palnadu District, Andhra Pradesh - 522 601.





### List of Experiments

**WEEK 1**  
Familiarization with programming environment


- Basic Linux environment and its editors like Vi, Vim & Emacs etc.
- Exposure to Turbo C, gcc
- Writing simple programs using printf(), scanf()

**WEEK 2**  
Converting algorithms/flow charts into C Source code.  
Developing the algorithms/flowcharts for the following sample programs


- Sum and average of 3 numbers
- Conversion of Fahrenheit to Celsius and vice versa
- Simple interest calculation

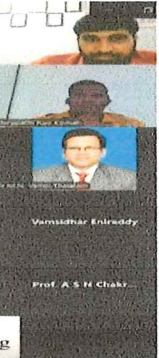
**WEEK 3**  
Simple computational problems using arithmetic expressions.

- Finding the square root of a given number
- Finding compound interest



**A.M. REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY**  
Approved by AICTE, New Delhi, Affiliated to JNTU - Kakinada, Accredited by NAAC  
*An Autonomous Institution*  
Web : [www.amreddyengineering.ac.in](http://www.amreddyengineering.ac.in), E-mail: [principal.amreddyengineering@gmail.com](mailto:principal.amreddyengineering@gmail.com) Ph : 98664 14252  
Vinukonda Road, Petalurivaripalem, Narasaraopet, Palnadu District, Andhra Pradesh - 522 601.



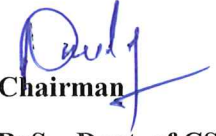


### SYLLABUS

**UNIT - I Introduction to Programming and Problem Solving**

History of Computers, Basic organization of a computer: ALU, input-output units, memory, program counter, Introduction to Programming Languages, Basics of a Computer Program Algorithms, flowcharts (Using Dia Tool), pseudo code, Introduction to Compilation and Execution, Primitive Data Types, Variables, and Constants, Basic Input and Output, Operation Type Conversion, and Casting.

Problem solving techniques: Algorithmic approach, characteristics of algorithm, Problem solving

  
**Chairman**  
 BoS – Dept. of CS

Copy to:

1. Principal
2. IQAC

## ANNEXURE -1

L	T	P	C
3	0	0	3

### AMR- 24

## INTRODUCTION TO PROGRAMMING

(Common to All branches of Engineering)

### Course Objectives:

- To introduce students to the fundamentals of computer programming.
- To provide hands-on experience with coding and debugging.
- To foster logical thinking and problem-solving skills using programming.
- To familiarize students with programming concepts such as data types, control structures, functions, and arrays.
- To encourage collaborative learning and teamwork in coding projects.

**Course Outcomes:** A student after completion of the course will be able to

**CO1:** Understand basics of computers, the concept of algorithm and algorithmic thinking.

**CO2:** Analyse a problem and develop an algorithm to solve it.

**CO3:** Implement various algorithms using the C programming language.

**CO4:** Understand more advanced features of C language.

**CO5:** Develop problem-solving skills and the ability to debug and optimize the code

### UNIT I

#### Introduction to Programming and Problem Solving

History of Computers, Basic organization of a computer: ALU, input-output units, memory, program counter, Introduction to Programming Languages, Basics of a Computer Program- Algorithms, flowcharts (Using Dia Tool), pseudo code. Introduction to Compilation and Execution, Primitive Data Types, Variables, and Constants, Basic Input and Output, Operations, Type Conversion, and Casting.

Problem solving techniques: Algorithmic approach, characteristics of algorithm, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms.

### UNIT II

#### Control Structures

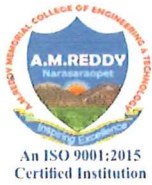
Simple sequential programs Conditional Statements (if, if-else, switch), Loops (for, while, do-while) Break and Continue.

### UNIT III

#### Arrays and Strings

Arrays indexing, memory model, programs with array of integers, two dimensional arrays, Introduction to Strings.





## UNIT IV

### Pointers & User Defined Data types

Pointers, dereferencing and address operators, pointer and address arithmetic, array manipulation using pointers, User-defined data types-Structures and Unions.

## UNIT V

### Functions & File Handling

Introduction to Functions, Function Declaration and Definition, Function call Return Types and Arguments, modifying parameters inside functions using pointers, arrays as parameters. Scope and Lifetime of Variables, Basics of File Handling

**Note:** The syllabus is designed with C Language as the fundamental language of implementation.

### Textbooks:

1. "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988
2. Schaum's Outline of Programming with C, Byron S Gottfried, McGraw-Hill Education, 1996

### Reference Books:

1. Computing fundamentals and C Programming, Balagurusamy, E., McGraw-Hill Education, 2008.
2. Programming in C, Rema Theraja, Oxford, 2016, 2<sup>nd</sup> edition
3. C Programming, A Problem Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE, 3<sup>rd</sup> edition



**ANNEXURE -2**

L	T	P	C
0	0	3	1.5

**AMR- 24**

**COMPUTER PROGRAMMING LAB**

**(Common to All branches of Engineering)**

**Course Objectives:**

The course aims to give students hands – on experience and train them on the concepts of the C- programming language.

**Course Outcomes:**

CO1: Read, understand, and trace the execution of programs written in C language. CO2: Select the right control structure for solving the problem.

CO3: Develop C programs which utilize memory efficiently using programming Constructs like pointers.

CO4: Develop Debug and Execute programs to demonstrate the applications of arrays, functions, basic concepts of pointers in C.

**UNIT - I**

**WEEK 1**

**Objective:** Getting familiar with the programming environment on the computer and writing the first program.

**Suggested Experiments/Activities:**

**Tutorial 1:** Problem-solving using Computers.

**Lab1:** Familiarization with programming environment

- i) Basic Linux environment and its editors like Vi, Vim & Emacs etc.
- ii) Exposure to Turbo C, gcc
- iii) Writing simple programs using printf(), scanf()

**WEEK 2**

**Objective:** Getting familiar with how to formally describe a solution to a problem in a series of finite steps both using textual notation and graphic notation.

**Suggested Experiments /Activities:**

**Tutorial 2:** Problem-solving using Algorithms and Flow charts.

**Lab 1:** Converting algorithms/flow charts into C Source code.

Developing the algorithms/flowcharts for the following sample programs

- i) Sum and average of 3 numbers
- ii) Conversion of Fahrenheit to Celsius and vice versa
- iii) Simple interest calculation



## WEEK 3

**Objective:** Learn how to define variables with the desired data-type, initialize them with appropriate values and how arithmetic operators can be used with variables and constants.

### **Suggested Experiments/Activities:**

**Tutorial 3:** Variable types and type conversions:

**Lab 3:** Simple computational problems using arithmetic expressions.

- i) Finding the square root of a given number
- ii) Finding compound interest
- iii) Area of a triangle using heron's formulae
- iv) Distance travelled by an object

## UNIT II

### WEEK 4

**Objective:** Explore the full scope of expressions, type-compatibility of variables & constants and operators used in the expression and how operator precedence works.

### **Suggested Experiments/Activities:**

**Tutorial4:** Operators and the precedence and as associativity:

**Lab4:** Simple computational problems using the operator' precedence and associativity

- i) Evaluate the following expressions.
  - a.  $A+B*C+(D*E) + F*G$
  - b.  $A/B*C-B+A *D/3$
  - c.  $A+++B---A$
  - d.  $J= (i++) + (++i)$
- ii) Find the maximum of three numbers using conditional operator
- iii) Take marks of 5 subjects in integers, and find the total, average in float

### WEEK 5

**Objective:** Explore the full scope of different variants of “if construct” namely if-else, null- else, if-else if\*-else, switch and nested-if including in what scenario each one of them can be used and how to use them. Explore all relational and logical operators while writing conditionals for “if construct”.

### **Suggested Experiments/Activities:**

**Tutorial 5:** Branching and logical expressions:

**Lab 5:** Problems involving if-then-else structures.

- i) Write a C program to find the max and min of four numbers using if-else.
- ii) Write a C program to generate electricity bill.
- iii) Find the roots of the quadratic equation.



- iv) Write a C program to simulate a calculator using switch case.
- v) Write a C program to find the given year is a leap year or not.

## WEEK 6

**Objective:** Explore the full scope of iterative constructs namely while loop, do-while loop and for loop in addition to structured jump constructs like break and continue including when each of these statements is more appropriate to use.

### Suggested Experiments/Activities:

**Tutorial 6:** Loops, while and for loops

**Lab 6:** Iterative problems e.g., the sum of series

- i) Find the factorial of given number using any loop.
- ii) Find the given number is a prime or not.
- iii) Compute sine and cos series
- iv) Checking a number palindrome
- v) Construct a pyramid of numbers.

## UNIT III WEEK 7:

**Objective:** Explore the full scope of Arrays construct namely defining and initializing 1-D and 2-D and more generically n-D arrays and referencing individual array elements from the defined array. Using integer 1-D arrays, explore search solution linear search.

### Suggested Experiments/Activities:

**Tutorial 7:** 1 D Arrays: searching.

**Lab 7:** 1D Array manipulation, linear search

- i) Find the min and max of a 1-D integer array.
- ii) Perform linear search on 1D array.
- iii) The reverse of a 1D integer array
- iv) Find 2's complement of the given binary number.
- v) Eliminate duplicate elements in an array.

## WEEK 8:

**Objective:** Explore the difference between other arrays and character arrays that can be used as Strings by using null character and get comfortable with string by doing experiments that will reverse a string and concatenate two strings. Explore sorting solution bubble sort using integer arrays.

### Suggested Experiments/Activities:

**Tutorial 8:** 2 D arrays, sorting and Strings.

**Lab 8:** Matrix problems, String operations, Bubble sort

- i) Addition of two matrices
- ii) Multiplication two matrices
- iii) Sort array elements using bubble sort
- iv) Concatenate two strings without built-in functions
- v) Reverse a string using built-in and without built-in string functions

## UNIT IV

### WEEK 9:

**Objective:** Explore pointers to manage a dynamic array of integers, including memory allocation & value initialization, resizing changing and reordering the contents of an array and memory de-allocation using malloc (), calloc (), realloc () and free () functions. Gain experience processing command-line arguments received by C.

#### Suggested Experiments/Activities:

**Tutorial 9:** Pointers, structures and dynamic memory allocation

**Lab 9:** Pointers and structures, memory dereference.

- i) Write a C program to find the sum of a 1D array using malloc()
- ii) Write a C program to find the total, average of n students using structures
- iii) Enter n students data using calloc() and display failed students list
- iv) Read student name and marks from the command line and display the student details alongwith the total.
- v) Write a C program to implement realloc()

### WEEK 10:

**Objective:** Experiment with C Structures, Unions, bit fields and self-referential structures (Singly linked lists) and nested structures

#### Suggested Experiments/Activities:

**Tutorial 10:** Bitfields, Self-Referential Structures, Linked lists

**Lab10 :** Bitfields, linked lists

Read and print a date using dd/mm/yyyy format using bit-fields and differentiate the samewithout using bit- fields

- i) Create and display a singly linked list using self-referential structure.
- ii) Demonstrate the differences between structures and unions using a C program.
- iii) Write a C program to shift/rotate using bitfields.
- iv) Write a C program to copy one structure variable to another structure of the same type.

### UNIT V WEEK 11:

**Objective:** Explore the Functions, sub-routines, scope and extent of variables, doing some experiments by parameter passing using call by value. Basic methods of numerical integration

#### Suggested Experiments/Activities:

**Tutorial 11:** Functions, call by value, scope and extent,

**Lab 11:** Simple functions using call by value, solving differential equations using Eulerstheorem.

- i) Write a C function to calculate NCR value.
- ii) Write a C function to find the length of a string.
- iii) Write a C function to transpose of a matrix.
- iv) Write a C function to demonstrate numerical integration of differential equations using Euler'smethod



## WEEK 12:

**Objective:** Explore how recursive solutions can be programmed by writing recursive functions that can be invoked from the main by programming at-least five distinct problems that have naturally recursive solutions.

### Suggested Experiments/Activities:

**Tutorial 12:** Recursion, the structure of recursive calls

### Lab 12:

 Recursive functions

- i) Write a recursive function to generate Fibonacci series.
- ii) Write a recursive function to find the lcm of two numbers.
- iii) Write a recursive function to find the factorial of a number.
- iv) Write a C Program to implement Ackermann function using recursion.
- v) Write a recursive function to find the sum of series.

## WEEK 13:

**Objective:** Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers

### Suggested Experiments/Activities:

**Tutorial 13:** Call by reference, dangling pointers

### Lab 13:

 Simple functions using Call by reference, Dangling pointers.

- i) Write a C program to swap two numbers using call by reference.
- ii) Demonstrate Dangling pointer problem using a C program.
- iii) Write a C program to copy one string into another using pointer.
- iv) Write a C program to find no of lowercase, uppercase, digits and other characters using pointers.

## WEEK 14:

**Objective:** To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.

### Suggested Experiments/Activities:

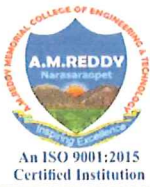
**Tutorial 14:** File handling

### Lab 14:

 File operations

- i) Write a C program to write and read text into a file.
- ii) Write a C program to write and read text into a binary file using fread() and fwrite()
- iii) Copy the contents of one file to another file.
- iv) Write a C program to merge two files into the third file using command-line arguments.
- v) Find no. of lines, words and characters in a file
- vi) Write a C program to print last n characters of a given file.





# A.M. REDDY MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi, Affiliated to JNTU - Kakinada, Accredited by NAAC

**An Autonomous Institution**

Web : [www.amreddyengineering.ac.in](http://www.amreddyengineering.ac.in), E-mail: [principal.amreddyengineering@gmail.com](mailto:principal.amreddyengineering@gmail.com) Ph : 98664 14252  
Vinukonda Road, Petturivaripalem, Narasaraopet, Palnadu District, Andhra Pradesh - 522 601.



## **Textbooks:**

1. Ajay Mittal, Programming in C: A practical approach, Pearson.
2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw Hill

## **Reference Books:**

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice-Hall of India
2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE

## ANNEXURE – 3

L	T	P	C
3	0	0	3

AMR – 24

### DATA STRUCTURES

(Common to CSE, IT & allied branches)

#### Course Objectives:

- To provide the knowledge of basic data structures and their implementations.
- To understand importance of data structures in context of writing efficient programs.
- To develop skills to apply appropriate data structures in problem solving.

#### Course Outcomes: At the end of the course, Student will be able to

**CO1:** Explain the role of linear data structures in organizing and accessing data efficiently in algorithms.

**CO2:** Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation.

**CO3:** Develop programs using stacks to handle recursive algorithms, manage program states, and solve related problems.

**CO4:** Apply queue-based algorithms for efficient task scheduling and breadth-first traversal in graphs and distinguish between deques and priority queues, and apply them appropriately to solve data management challenges.

**CO5:** Devise novel solutions to small scale programming challenges involving data structures such as stacks, queues, Trees.

**CO6:** Recognize scenarios where hashing is advantageous, and design hash-based solutions for specific problems.

#### UNIT I

**Introduction to Linear Data Structures:** Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity analysis for linear data structures. Searching Techniques: Linear & Binary Search, Sorting Techniques: Bubble sort, Selection sort, Insertion Sort

#### UNIT II

**Linked Lists:** Singly linked lists: representation and operations, doubly linked lists and circular linked lists, Comparing arrays and linked lists, Applications of linked lists.

#### UNIT III

**Stacks:** Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.

#### UNIT IV

**Queues:** Introduction to queues: properties and operations, implementing queues



using arrays and linked lists, Applications of queues in breadth-first search, scheduling, etc.

**Deque:** Introduction to deques (double-ended queues), Operations on deques and their applications.

## UNIT V

Trees: Introduction to Trees, Binary Search Tree – Insertion, Deletion & Traversal

**Hashing:** Brief introduction to hashing and hash functions, Collision resolution techniques: chaining and open addressing, Hash tables: basic implementation and operations, Applications of hashing in unique identifier generation, caching, etc.

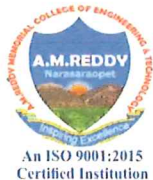
### Textbooks:

1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2<sup>nd</sup> Edition.
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Press, 2008

### Reference Books:

1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders
2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft
3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum
4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms" by Robert Sedgwick





## ANNEXURE – 4

### AMR -24

### DATA STRUCTURES LAB

(Common to CSE, IT & allied branches)

L	T	P	C
0	0	3	1.5

#### Course Objectives:

The course aims to strengthen the ability of the students to identify and apply the suitable data structure for the given real-world problem. It enables them to gain knowledge in practical applications of data structures.

**Course Outcomes:** At the end of the course, Student will be able to

**CO1:** Explain the role of linear data structures in organizing and accessing data efficiently in algorithms.

**CO2:** Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation.

**CO3:** Develop programs using stacks to handle recursive algorithms, manage program states, and solve related problems.

**CO4:** Apply queue-based algorithms for efficient task scheduling and breadth-first traversal in graphs and distinguish between deques and priority queues and apply them appropriately to solve data management challenges.

**CO5:** Recognize scenarios where hashing is advantageous, and design hash-based solutions for specific problems.

#### List of Experiments:

##### Exercise 1: Array Manipulation

- Write a program to reverse an array.
- C Programs to implement the Searching Techniques – Linear & Binary Search
- C Programs to implement Sorting Techniques – Bubble, Selection and Insertion Sort

##### Exercise 2: Linked List Implementation

- Implement a singly linked list and perform insertion and deletion operations.
- Develop a program to reverse a linked list iteratively and recursively.
- Solve problems involving linked list traversal and manipulation.

##### Exercise 3: Linked List Applications

- Create a program to detect and remove duplicates from a linked list.
- Implement a linked list to represent polynomials and perform addition.
- Implement a double-ended queue (deque) with essential operations.

##### Exercise 4: Double Linked List Implementation

- Implement a doubly linked list and perform various operations to understand its properties and applications.
- Implement a circular linked list and perform insertion, deletion, and traversal.



### Exercise 5: Stack Operations

- i) Implement a stack using arrays and linked lists.
- ii) Write a program to evaluate a postfix expression using a stack.
- iii) Implement a program to check for balanced parentheses using a stack.

### Exercise 6: Queue Operations

- i) Implement a queue using arrays and linked lists.
- ii) Develop a program to simulate a simple printer queue system.
- iii) Solve problems involving circular queues.

### Exercise 7: Stack and Queue Applications

- i) Use a stack to evaluate an infix expression and convert it to postfix.
- ii) Create a program to determine whether a given string is a palindrome or not.
- iii) Implement a stack or queue to perform comparison and check for symmetry.

### Exercise 8: Binary Search Tree

- i) Implementing a BST using Linked List.
- ii) Traversing of BST.

### Exercise 9: Hashing

- i) Implement a hash table with collision resolution techniques.
- ii) Write a program to implement a simple cache using hashing.

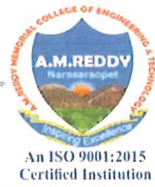
### Textbooks:

1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2<sup>nd</sup> Edition.
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Press, 2008

### Reference Books:

1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders
2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft
3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum
4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms by Robert Sedgewick





## ANNEXURE – 5

L	T	P	C
0	0	2	1

### AMR – 24

## ITWORKSHOP

(Common to all branches of Engineering)

### Course Objectives:

- To introduce the internal parts of a computer, peripherals, I/O ports, connecting cables
- To demonstrate configuring the system as Dual boot both Windows and other Operating Systems Viz. Linux, BOSS
- To teach basic command line interface commands on Linux.
- To teach the usage of Internet for productivity and self-paced life-long learning
- To introduce Compression, Multimedia and Antivirus tools and Office Tools such as Word processors, Spread sheets and Presentation tools.

### Course Outcomes:

- CO1:** Perform Hardware troubleshooting.
- CO2:** Understand Hardware components and inter dependencies.
- CO3:** Safeguard computer systems from viruses/worms.
- CO4:** Document/ Presentation preparation.
- CO5:** Perform calculations using spreadsheets.

### PC Hardware & Software Installation

**Task 1:** Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

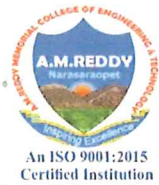
**Task 2:** Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

**Task 3:** Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

**Task 4:** Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot (VMWare) with both Windows and Linux. Lab instructors should verify the installation and follow it up with a Viva

**Task 5:** Every student should install BOSS on the computer. The system should be configured as dual boot (VMWare) with both Windows and BOSS. Lab instructors should verify the installation and follow it up with a Viva





## Internet & World Wide Web

**Task1:** Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Task 2:** Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

**Task 3:** Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

**Task 4:** Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

## LaTeX and WORD

**Task 1 – Word Orientation:** The mentor needs to give an overview of La TeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of La TeX and MS office or equivalent (FOSS) tool Word as word Processors, Details of the four tasks and features that would be covered in each, Using La TeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

**Task 2:** Using La TeX and Word to create a project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both La TeX and Word.

**Task 3:** Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

**Task 4:** Creating a Newsletter: Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

## EXCEL

**Excel Orientation:** The mentor needs to tell the importance of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.



**Task 1:** Creating a Scheduler - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

**Task 2:** Calculating GPA -. Features to be covered:- Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function

## LOOKUP/VLOOKUP

**Task 3:** Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting

## POWER POINT

**Task 1:** Students will be working on basic power point utilities and tools which help them create basic power point presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint.

**Task 2:** Interactive presentations - Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

**Task 3:** Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

## AI TOOLS – ChatGPT

**Task 1:** Prompt Engineering: Experiment with different types of prompts to see how the model responds. Try asking questions, starting conversations, or even providing incomplete sentences to see how the model completes them.

- Ex: Prompt: "You are a knowledgeable AI. Please answer the following question: What is the capital of France?"

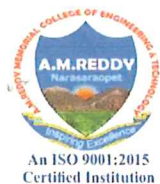
**Task 2:** Creative Writing: Use the model as a writing assistant. Provide the beginning of a story or a description of a scene, and let the model generate the rest of the content. This can be a fun way to brainstorm creative ideas

- Ex: Prompt: "In a world where gravity suddenly stopped working, people started floating upwards. Write a story about how society adapted to this new reality."

**Task 3:** Language Translation: Experiment with translation tasks by providing a sentence in one language and asking the model to translate it into another language. Compare the output to see how accurate and fluent the translations are.

- Ex: Prompt: "Translate the following English sentence to French: 'Hello, how are you doing today?'"





## Reference Books:

2. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dream tech, 2003
3. The Complete Computer upgrade and repair book, Cheryl A Schmidt, WILEY Dream tech, 2013, 3rd edition
4. Introduction to Information Technology, IITL Education Solutions limited, Pearson Education, 2012, 2<sup>nd</sup> edition
5. PC Hardware - A Handbook, Kate J. Chase, PHI (Microsoft)
6. LaTeX Companion, Leslie Lamport, PHI/Pearson.
7. IT Essentials PC Hardware and Software Companion Guide, David Anfinson and Ken Quamme. – CISCO Press, Pearson Education, 3<sup>rd</sup> edition
8. IT Essentials PC Hardware and Software Labs and Study Guide, Patrick Regan– CISCO Press, Pearson Education, 3<sup>rd</sup> edition

