NEP

Syllabus for

B.Sc. Computer Science

Major,

Minor

and

Interdisciplinary

Semester	**level**		Major	Lab	Minor	Lab	Multidisciplinary	Lab
I	100	1	Computer Fundamentals and programming with C	Y	Programming Fundamentals using C	Y	Programming Fundamentals using C	Y
		2	Basic Electronics with Digital Logic	Y				
II		3	Data Structures	Y			Data Structures using C++	Y
		4	Operating Systems	Y				
III	200	5	Object Oriented Programing	Y	Data Structures using C++	Y	Operating Systems	Y
		6	Database Management System	Y				
		7	Computational Mathematics	N				
IV		8	Analysis of Algorithms	Y			Database Management System	Y
		9	Computer System Architecture	N				
		10	Networking	N				
V	300	11	Software Engineering	Y	Operating Systems	Y	Computer System Architecture	N
		12	Graphics and Multimedia	Y				
VI		13	Theory of Computation	N			Computer Networks	N
		14	Microprocessor & VLSI Design	Y				
VII	400	15	Artificial Intelligence	Y	Database Management System	Y		
		16	Compiler Design	N				
		17	Minor Project	LAB				
VIII		18	Image Processing and Computer Vision	Y				
		19	Machine Learning	Y				
		20	Big Data	N				
		21	Cyber Security and IoT	N				

CORE COURSES (HONOURS IN COMPUTER SCIENCE)

CMSACOR01T: Computer Fundamentals and programming with C **Theory: 45 Lectures**

Overview of C **(5 Lectures)**

History, Basic Structure, Algorithms, Structured programming constructs. Character sets, Tokens, Keywords, Constants, Variables, Data Types, Declaration of storage classes.

Operators, Expressions and Preprocessor **(8 Lectures)**

Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional, Bitwise, Special operator, Operator Precedence and Associativity; Arithmetic Expressions, Evaluation of expression, type casting. Comments, Input and output operations. Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

Decision and Loop Control Structure **(7 Lectures)**

If-else statements, Nested if-else, switch, Conditional operator. While, do-While, for loop, break statements, continue statements, goto statements.

Functions and Arrays **(7 Lectures)**

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays, return statement, return values and their types, String handling with arrays, String handling functions, recursion

Pointers **(6 Lectures)**

Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation.

User defined Datatypes and Memory Allocation **(6 Lectures)**

Enumerated datatypes, Structures. Structure arrays, Pointers to Functions and Structures, Unions. Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation

File Access **(6 Lectures)**

Opening and closing a file (use of fstream header file, ifstream, ofstream), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files,

Text Books

1. Let Us C, Kanetkar, BPB Publication.
2. Programming in ANSI C, Balaguruswamy, McGraw Hill.
3. Programming with C, Byron S. Gottfried, McGraw Hill.

Reference Books

1. The C Programming Language, Kernighan and Dennis Ritchie, PHI.
2. The Complete reference C, Herbert Schildt, McGraw Hill.
3. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

CMSACOR01P: Computer Fundamentals and programming with C Lab Practical: 60 Lectures

Instruction: Use an open source C compiler.

1. Write a program (WAP) to print the sum and product of digits of an integer.
2. WAP to reverse a non-negative integer.
3. WAP to compute the sum of the first n terms of the following series
   S = 1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first n terms of the following series,
   S =1-2+3- 4+5…………….
5. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
6. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by the user is Palindrome or not.
7. WAP to compute the factors of a given number.
8. WAP to swap two numbers using macro.
9. WAP to print a triangle of stars as follows (take number of lines from user):

```
    *
   ***
  *****
 *******
*********
```

10. WAP to perform following actions on an array entered by the user :
    a. Print the even-valued elements
    b. Print the odd-valued elements
    c. Calculate and print the sum and average of the elements of array
    d. Print the maximum and minimum element of array
    e. Remove the duplicates from the array
    f. Print the array in reverse order

    (The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.)

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function passes the address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find the sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operators.
16. Write a menu driven program to perform following operations on strings:
    a. Show address of each character in string
    b. Concatenate two strings without using strcat function.
    c. Concatenate two strings using strcat function.
    d. Compare two strings
    e. Calculate length of the string (use pointers)
    f. Convert all lowercase characters to uppercase
    g. Convert all uppercase characters to lowercase
    h. Calculate number of vowels
    i. Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from the user and store them in an array. Implement an array using pointers. The program will print the array elements in ascending and descending order.
25. Add two distances in a meter kilometer system using structure.
26. Add two complex numbers using structures.
27. Calculate the difference between two time periods using structures.

CMSACOR02T: Basic Electronics with Digital Logic Lab **Theory: 45 Lectures**

Elementary Physics of semiconductors **(5 Lectures)**

Intrinsic and Extrinsic semiconductors, P & N type, Diode & its applications: P-N Junction diodes, Biasing of a junction diode, Depletion region & its effect, Diode as a rectifier, Types of diodes, LED, LCD.

Principle of junction transistors **(7 Lectures)**

Current components of a transistor, Modes of a transistor (CB, CE and CC) and their properties, I/O characteristics of a transistor in CE mode. Relation between parameters of Transistor, Biasing of a transistor: Q point, load line, Self-bias, fixed bias & collector to base bias. Amplifiers: Concepts, Brief idea of Class A & B.

Inverters using Transistors–transfer characteristics and threshold voltages.

Combinational Circuits **(8 Lectures)**

Realization of AND and OR Gates using diodes and NOT Gate using transistors, Standard Gate Assemblies, IC chips packaging nomenclature, Half and Full Adder(3 & bit), Multi-bit adders – Ripple carry and Carry Look Ahead Adder, Adder/subtractor, BCD-Adder.

Data selectors/multiplexers: **(8 Lectures)**

Expansions, reductions, function realization, universal function realization, multi-function realization, Decoders: function realization, Demultiplexer and function realization, Encoder, Priority Encoder, Parity bit Generator/checker, Gray Code Generator, Code Converters.

Sequential Circuits **(9 Lectures)**

Model of Sequential computing, Difference between Combinational and Sequential circuit, RSLatch: using NAND and NOR Gates, RS Latch as a Static RAM Cell, Problems of Basic Latch circuits, Digital Clock – Duty Cycle, Rising time, Falling time, Clocked Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flip-flops, Flip-flops with Preset and Clear, Application of Flip-flops: Asynchronous Counter(UP/DOWN) upto 4 bit counter, Decade Counter, Mod – n Counter, Synchronous Counters – different mod counters, Ring counter, Registers, Shift Registers.

Data Converter **(8 Lectures)**

D/A Conversion principle using basic circuit, R-2R Ladder circuit, Counter based A/D converter, Successive approximation method for A/D conversion. DTL and TTL NAND gate circuits and its operations, Fan in & Fan out. SSI, MSI, LSI, and VLSI classifications.

Text Books

1. Introduction to Computer Science by P. K. Sinha, PHI
2. Electronics Fundamentals and Applications by D. Chattopadhay and P. C. Rakshit, 6th Edition, New Age International (P)
3. Digital Circuits and Design by S. Salivahanan, 5th Edition, Oxford University Press
4. Digital Logic and Computer Design by Morris Mano, PHI
5. Digital Principle and Applications by Malvino & Leach, TMH
6. Computer System Architecture by M. Morris Mano

Reference Books

1. Digital Systems Principles and Applications by Ronal J. Tocci and Neal S. Widmer, 8th Edition, PHI
2. Digital Fundamentals by Floyd, Pearson Education
3. Electronics Devices and Circuit Theory by Boylestad, Nashelsky, PHI

CMSACOR02P: Basic Electronics with Digital Logic Lab **Practical: 60 Lectures**

1. Implement Half Adder/Half Subtractor/Full Adder/Full Subtractor using Logic Gates. Realize a logic function using basic/universal gates in SOP and POS form. Study the functionalities of 7483 and design a BCD adder using 7483 or equivalent.
2. Design of two level AND – OR, NAND –NAND, NOR-NOR circuits to realize any truth table. Realize XOR in two level and multilevel.
3. Design a 4 bit 2's complement adder – subtractor unit using 7483 or equivalent and XOR gates.
4. Design a circuit to convert BCD numbers to corresponding gray codes.
5. Design a 4:1 MUX using NAND gates. Study of 74153 and 74151. Design Full Adder/Subtractor using MUX.
6. Design a 2:4 decoder using NAND gates. Study of 74155 and 74138. Design Full Adder/Subtractor using decoders.
7. Design a parity generator/checker using basic gates.
8. Design magnitude comparator using basic/universal gates. Study of 7485.
9. Design a seven segment display unit.
10. Realize S-R, D, J-K and T flip-flop using basic gates. (Study the undefined state in S-R flip-flop).
11. Design a shift register (shift left and shift right) using flip-flops. (Study the functional characteristic of IC74194 with emphasis on timing diagram).
12. Design Asynchronous and Synchronous counters. Study of IC 74193.

CMSACOR03T: Data Structures **Theory: 45 Lectures**

Introduction **(5 Lectures)**

Data Object, Abstract Data Type, Data Structures and Data Types. Types of Data Structures – Linear and non-linear Data Structures.Single and Multi-dimensional Arrays, Address Calculations, Sparse Matrices (Array Representation).

Linked Lists **(7 Lectures)**

Singly, Doubly and Circular Lists (Array and Linked representation); Operations on Lists. Sparse Matrices (Linked Representation).

Stacks and Queues **(9 Lectures)**

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack. Array and Linked representation of Queue, De-queue, Priority Queues

Recursion **(5 lectures)**

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Binary Trees **(10 Lectures)**

Introduction; Properties, Binary Trees Traversals (Recursive and Non-Recursive), Binary Search Trees (Insertion, Deletion), Recursive and Iterative Traversals on Binary Search Trees; Threaded Binary Trees (Concept only); Height-Balanced Trees (Concept only).

Searching, Sorting and Hashing **(9 Lectures)**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Bubble

Sort, Comparison of Sorting Techniques. Introduction to Hashing, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining and simple examples.

Text Books

1. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.
2. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.

Reference Books

1. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.
2. D. S. Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
3. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
4. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java", 2003.
5. Samanta, D. "Classic data structures.", PHI, *Terminology* 2 (2001): 1.
6. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.

CMSACOR03P: Data Structures Lab **Practical: 60 Lectures**

Instruction: Use an open source C++ compiler.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.
2. WAP to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomials.
11. WAP to calculate factorial and to compute the factors of a given no. (i) using recursion, (ii) using iteration
12. WAP to display Fibonacci series (i) using recursion, (ii) using iteration
13. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion

14. WAP to create a Binary Search Tree and include following operations in tree:
    i. Insertion (Recursive and Iterative Implementation)
    ii. Deletion by copying
    iii. Deletion by Merging
    iv. Search a no. in BST
    v. Display its preorder, postorder and inorder traversals Recursively
    vi. Display its preorder, postorder and inorder traversals Iteratively
    vii. Display its level-by-level traversals
    viii. Count the non-leaf nodes and leaf nodes
    ix. Display height of tree
    x. Create a mirror image of tree
    xi. Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.

CMSACOR04T: Operating Systems Theory: 45 Lectures

Introduction (7 Lectures)

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

Operating System Organization (6 Lectures)

Processor and user modes, kernels, system calls and system programs.

Process Management (16 Lectures)

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

Memory Management (8 Lectures)

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

File and I/O Management (8 Lectures)

Directory structure, file operations, file allocation methods, device management.

Text Books

1. Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A. S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.

Reference Books

1. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
2. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.
3. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

CMSACOR04P: Operating Systems Lab **Practical: 60 Lectures**

Group I: C Programs (Use an open source C compiler.)

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
   a. same program, same code.
   b. same program, different code.
   c. before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.

Group II: Shell Programs (Use an open source Linux compiler.)

1. Write a Shell Script to determine all prime numbers between two positive integers a and b. a and b are supplied as command line arguments.
2. Write a Shell Script that counts the number of words in a line of text. The line text will be provided as command line arguments.
3. Write a Shell Script/Program to check whether a given string is a Palindrome or not.
4. Write a Shell Script/Program to print the command line arguments in reverse order.
5. Write a Shell Script/Program to modify the cal command of UNIX to include the following.

i) Print the current month and current year when called without parameter.

ii) Print the calendar of the month of the current year if only one argument between 1 and 12.

Minor / Multidisciplinary

CMSGCOR01T: Programming Fundamentals using C **Theory: 45 Lectures**

Introduction to Computers **(6 Lectures)**

Characteristics of Computers, Uses of computers, Types and generations of Computers. Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices, number system, logic gates.

Planning the Computer Program **(6 Lectures)**

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. Flow charting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

Overview of C & Programming elements **(5 Lectures)**

History, Basic Structure, Algorithms, Structured programming constructs. Character sets, C Token, Keywords, Constants, Variables, Data Types, Declaration of storage classes.

Operators, Expressions and Preprocessor **(8 Lectures)**

Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional, Bitwise, Special operator, Operator Precedence and Associativity; Arithmetic Expressions,Evaluation of expression, type casting. Comments, Input and output operations. Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

Decision & Loop Control Structure **(7 Lectures)**

If-else statements, Nested if-else, switch, Conditional operator. While, do-While, for loop, break statements, continue statements, goto statements.

Functions and Arrays **(7 Lectures)**

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays, return statement, return values and their types, String handling with arrays, String handling functions, recursion

Pointers **(6 Lectures)**

Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation.

Text Books

1. Let Us C, Kanetkar, BPB Publication.
2. Programming in ANSI C, Balaguruswamy, McGraw Hill.
3. Programming with C, Byron S. Gottfried, McGraw Hill.

Reference Books

1. The C Programming Language, Kernighan and Dennis Ritchie, PHI.
2. The Complete reference C, Herbert Schildt, McGraw Hill.
3. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

CMSACOR01P: Programming Fundamentals using C Lab **Practical: 60 Lectures**

Instruction: Use an open source C compiler.

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a non-negative integer.
3. WAP to compute the sum of the first n terms of the following series
4. S=1+1/2+1/3+1/4+……
5. WAP to compute the sum of the first n terms of the following series, S =1-2+3- 4+5…………….
6. Write a function to find whether a given no. is prime or not. Use the same to generate the
7. prime numbers less than 100.
8. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by the user is Palindrome or not.
9. WAP to compute the factors of a given number.
10. WAP to swap two numbers using macro.
11. WAP to print a triangle of stars as follows (take number of lines from user):
12. *
13. ***
14. *****
15. *******
16. *********
17. WAP to perform following actions on an array entered by the user :
18. Print the even-valued elements
19. Print the odd-valued elements
20. Calculate and print the sum and average of the elements of array
21. Print the maximum and minimum element of array
22. Remove the duplicates from the array
23. Print the array in reverse order
24. The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
25. WAP that prints a table indicating the number of occurrences of each alphabet in the text

entered as command line arguments.

26. Write a program that swaps two numbers using pointers.
27. Write a program in which a function passes the address of two variables and then alter its contents.
28. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
29. Concatenate two strings without using strcat function.
30. Concatenate two strings using strcat function.
31. Compare two strings
32. Calculate length of the string (use pointers)
33. Convert all lowercase characters to uppercase
34. Convert all uppercase characters to lowercase
35. Calculate number of vowels
36. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
37. WAP to display Fibonacci series (i) using recursion, (ii) using iteration.
38. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration.
39. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
40. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
41. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
42. Write a program that will read 10 integers from the user and store them in an array. Implement an array using pointers. The program will print the array elements in ascending and descending order.
43. Add two distances in a meter kilometer system using structure.
44. Add two complex numbers using structures.

CMSGCOR02T: Data Structures Using C++

Theory: 45 Lectures

Introduction **(5 Lectures)**

Data Object, Abstract Data Type, Data Structures and Data Types. Types of Data Structures – Linear and non-linear Data Structures.Single and Multi-dimensional Arrays, Address Calculations, Sparse Matrices (Array Representation).

Linked Lists **(7 Lectures)**

Singly, Doubly and Circular Lists (Array and Linked representation); Operations on Lists. Sparse Matrices (Linked Representation).

Stacks and Queues **(9 Lectures)**

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack. Array and Linked representation of Queue, De-queue, Priority Queues

Recursion **(5 lectures)**

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of

Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Binary Trees **(10 Lectures)**

Introduction; Properties, Binary Trees Traversals (Recursive and Non-Recursive), Binary Search Trees (Insertion, Deletion), Recursive and Iterative Traversals on Binary Search Trees; Threaded Binary Trees (Concept only); Height-Balanced Trees (Concept only).

Searching, Sorting and Hashing **(9 Lectures)**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Comparison of Sorting Techniques. Introduction to Hashing, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining and simple examples.

Text Books

1. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.
2. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.

Reference Books

1. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.
2. D.S Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
3. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
4. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java", 2003.
5. Samanta, D. "Classic data structures.", PHI, *Terminology* 2 (2001): 1.
6. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
7. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009
8. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley,2013
9. Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
10. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.
11. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.

CMSACOR03P: Data Structures Lab (Using C++) **Practicals: 60 Lectures**

Instruction: Use an open source C++ compiler.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.
2. WAP to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List. Include functions for insertion, deletion and search of a

number, reverse the list.

5. Implement Circular Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomials.
11. WAP to create a Binary Search Tree and include following operations in tree:
    - Insertion (Recursive and Iterative Implementation)
    - Deletion by copying
    - Deletion by Merging
    - Search a no. in BST
    - Display its preorder, postorder and inorder traversals Recursively
    - Display its preorder, postorder and inorder traversals Iteratively
    - Display its level-by-level traversals
    - Count the non-leaf nodes and leaf nodes
    - Display height of tree
    - Create a mirror image of tree
    - Check whether two BSTs are equal or not
12. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
13. WAP to reverse the order of the elements in the stack using additional stack.