

Course Name : Computer Engineering Group **Course code:** CO/CM/IF/CD

Semester : Fourth

Subject Title : Data structure

Subject Code: 9067

Teaching and Examination Scheme:

Teaching Scheme			Examination Scheme					
TH	TU	PR	TH	TEST	PR	OR	TW	TOTAL
03	--	04	80	20	50#	--	25@	175

Rationale:

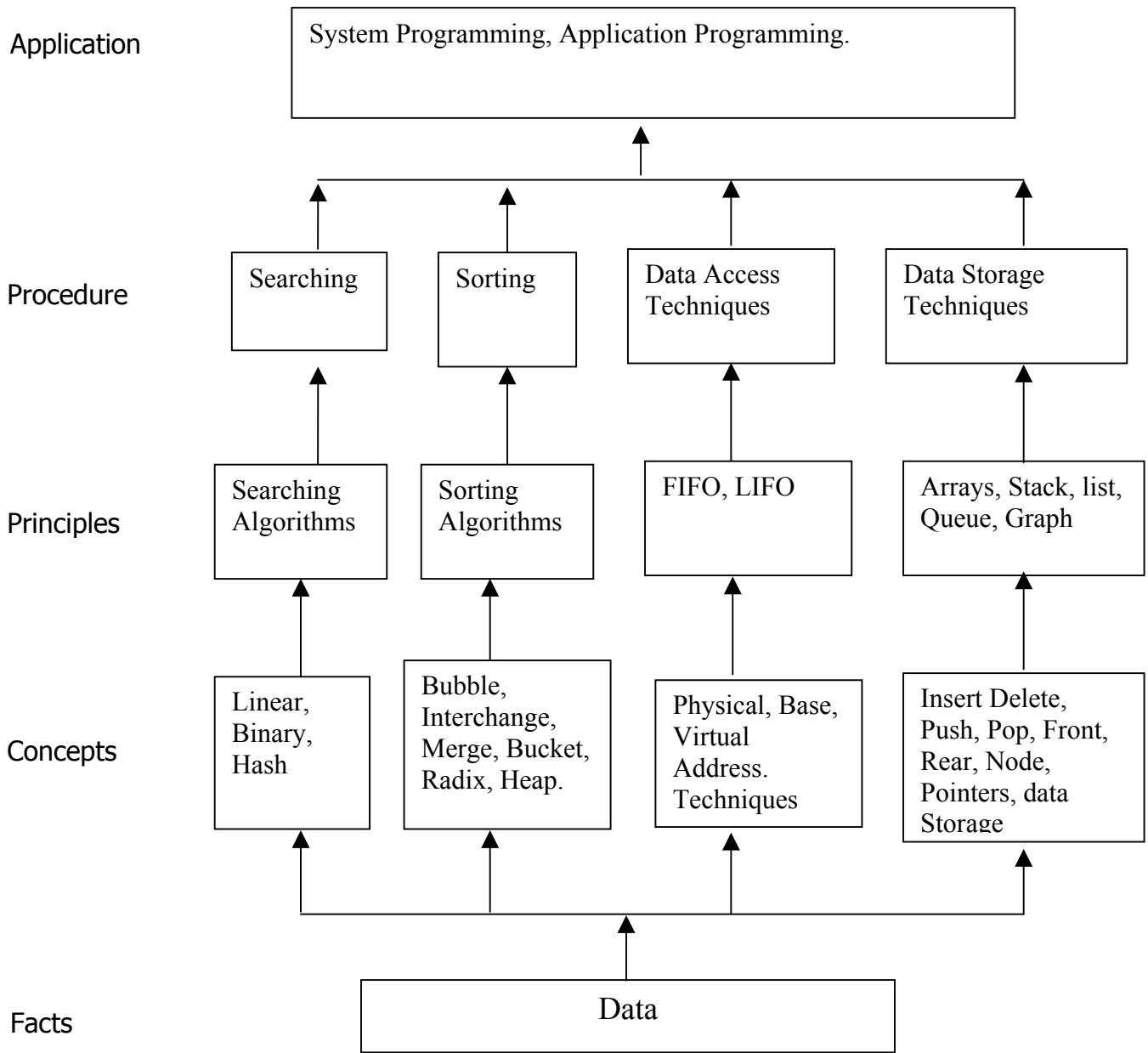
The primary objective of this course is to provide the student with an advanced treatment of computer programming with an emphasis on design and implementation of abstract data structures. The coding language is C/C++.

Objectives:

Students will able to:

1. Write complex applications using structured programming methods.
2. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and trees.
3. Use various data structures effectively in application programs.
4. Implement various data structures in more than one manner.
5. Compare different implementations of data structures and to recognize the advantages and disadvantages of the different implementations.
6. Demonstrate understanding of various sorting algorithms, including bubble sort, insertion sort, selection sort, heap sort and quick-sort. C
7. Compare the efficiency of various sorting algorithms in terms of both time and space.
8. Program multiple file programs in a manner that allows for reusability of code.
9. Trace and code recursive functions.

Learning Structure:



Contents: Theory

Chapter	Name of the Topic	Hours	Marks
01	Introduction to data structure: 1.1 Data Representation 1.2 Abstract data Types 1.3 Data Structure and Structured Types 1.4 Atomic Type 1.5 Difference between Abstract Data Types, Data Types And Data Structures 1.6Data Types 1.7Linear data type 1.8Non- Linear data type 1.9Primitive data type 1.10Non primitive data type 1.11Refinement Stages	04	06
02	Principles of programming and Analysis of Algorithms: 2.1Algorithms 2.2Different approaches for designing an algorithm 2.3Complexity 2.4Big 'O' Notation 2.5Algorithm analysis	02	06
03	Searching & Sorting: 3.1 Sorting-An Introduction 3.2 Efficiency of Sorting Algorithms 3.3 Bubble Sort 3.4 Selection Sort 3.5 Quick Sort 3.6 Insertion Sort 3.7 Merge Sort 3.8 Binary Tree Sort 3.9 Radix Sort 3.10 Shell Sort 3.11 Heap Sort 3.12 Searching-An Introduction 3.13 Binary Search	08	15
04	Stacks: 4.1 Introduction to Stacks 4.2 Stacks as an Abstract Data Type 4.3Primitive operations of stacks 4.3 Representation of Stacks through Arrays 4.4 Representation of Stacks through Linked List 4.5 Application of Stacks	06	10

	4.6 Stack and Recursion		
05	Queues: 5.1 Introduction 5.2 Queue as an Abstract Data Type 5.3 Representation of Queues 5.4 Operations on queue : Searching ,Insertion, Deletion. 5.5 Circular Queues 5.6 Priority Queue 5.7 Application of Queues	06	10
06	Linked List: 6.1 introduction, 6.2 Terminologies Node, Address, Pointer, Information, Next, Null pointer, Empty list etc. 6.3 Operations on list Searching, Insertion and Deletion 6.4 Types of lists Linked list and Circular list 6.5 Array stacks, queues, implementation using list.	06	08
07	Trees: 7.1 Introduction to Binary Trees 7.2 Types of Trees 7.3 Basic Definition of Binary Trees 7.4 Operations on Binary Search Tree 7.5 Type of tree Binary, Height balanced and Weight balanced tree 7.4 Operations on trees, 7.5. Searching Depth-first search and Breadth-first search 7.5 Traversing Pre-order, In-order and Post-order 7.6 Insertion, 7.7 Deletion,	08	12
08	Graphs: 8.1 Introduction to Graphs 8.2 Terms Associated with Graphs 8.3 Terminology graph, node (vertices), arcs (edge), directed graph, in-degree, out-degree, adjacent, successor, predecessor, relation, Weight, path, length 8.4 Sequential Representation of Graphs 8.5 Linked Representation of Graphs 8.6 Traversal of Graphs 8.7 Spanning Trees 8.8 Shortest Path 8.9 Application of Graph	06	08

09	Hashing 9.1 Hash functions 9.2 Deleting items from hash tables	02	05
	Total	48	80

Practical:

Skills to be developed:

Intellectual skills:

- Use of programming language constructs in program implementation.
- To be able to apply different logics to solve given problem.
- To be able to write program using different implementations for the same problem
- Study different types of errors as syntax semantic, fatal, linker & logical
- Debugging of programs
- Understanding different steps to develop program such as
 - Problem definition
 - Analysis
 - Design of logic
 - Coding
 - Testing
 - Maintenance (Modifications, error corrections, making changes etc.)

Motor skills:

- Proper handling of Computer System.

List of Practical:

Sr.No.	Practical
01	Programs based on: Array operations, insertion, deletion
02	Programs for implementing various sorting techniques. (Minimum three sorting techniques from topics mentioned in the syllabus))
03	Programs for implementing various sorting and searching techniques. (Minimum two searching techniques from topics mentioned in the syllabus.)
04	Programs based on Stacks Implementation of PUSH & POP operations, Evaluate postfix expressions, Infix to postfix conversions.
05	Recursive programs: factorial, fibonacci, Ackerman function, and tower of

	Hanoi.(any two)
06	Programs for demonstrating queue operations. one recursive program converted to non recursive ones
07	Two programs based on Linked lists
08	Programs based on trees Creating a binary tree, in order, preorder and post order traversal of binary tree, deleting a node from binary tree.
09	Assignments based on graph theory.
10	Program based on hashing

Learning Resources:

Books:

Sr. No.	Author	Title	Publisher
01	ISRD Group New Dehli	Data Structure Using C	Tata Magraw Hill
02	Tremblie and Sorrenson	Data Structures	TMH Publications
03	Lafore	Teach Yourself data Structure and Algorithms in 24 Hrs.	BPB Publication