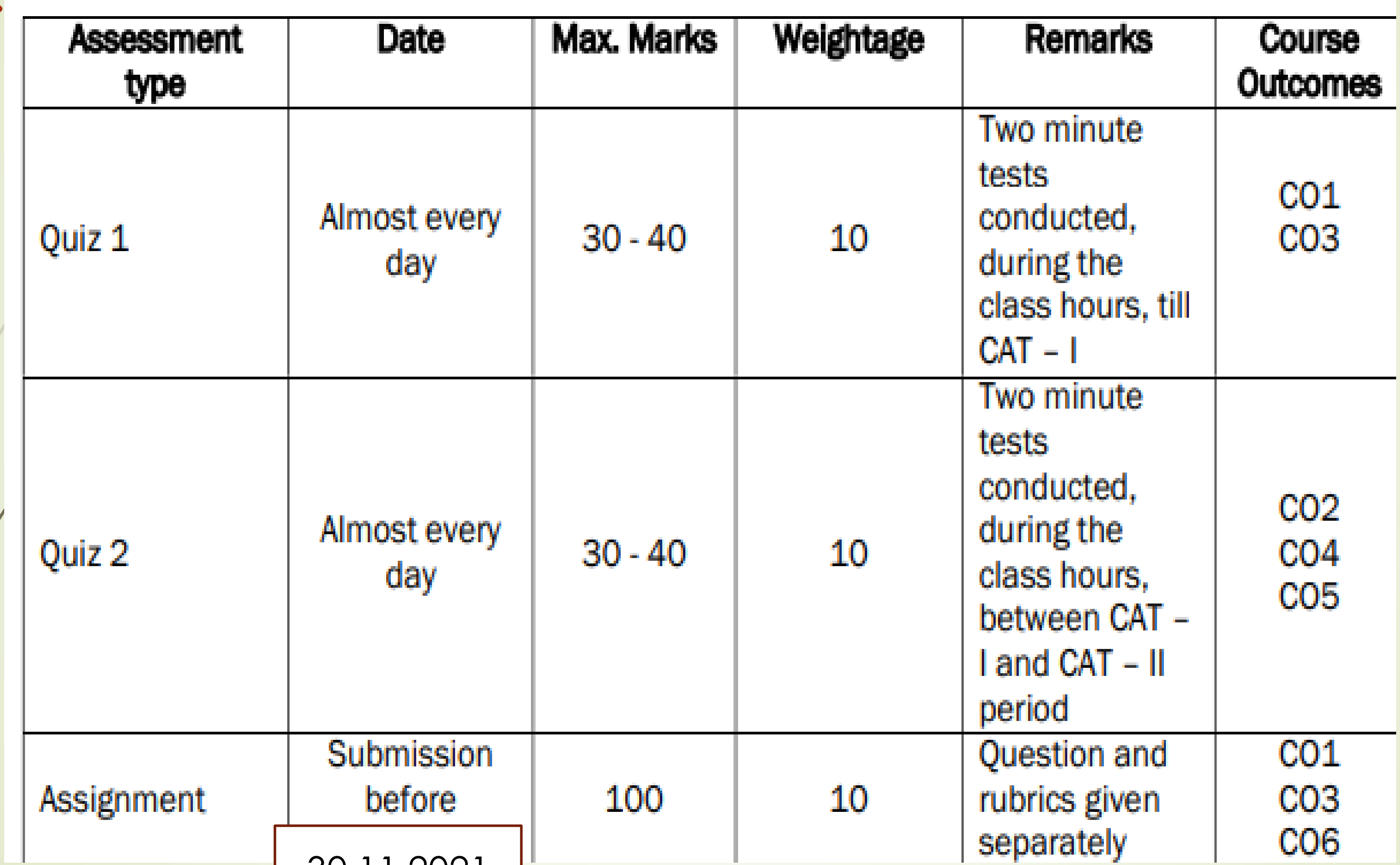
Data Structure and Algorithms

Dr. Subhra Rani Patra
SCOPE, VIT Chennai



Assessment type	Date	Max. Marks	Weightage	Remarks	Course Outcomes
Quiz 1	Almost every day	30 - 40	10	Two minute tests conducted, during the class hours, till CAT – I	CO1 CO3
Quiz 2	Almost every day	30 - 40	10	Two minute tests conducted, during the class hours, between CAT – I and CAT – II period	CO2 CO4 CO5
Assignment	Submission before 30.11.2021	100	10	Question and rubrics given separately	CO1 CO3 CO6

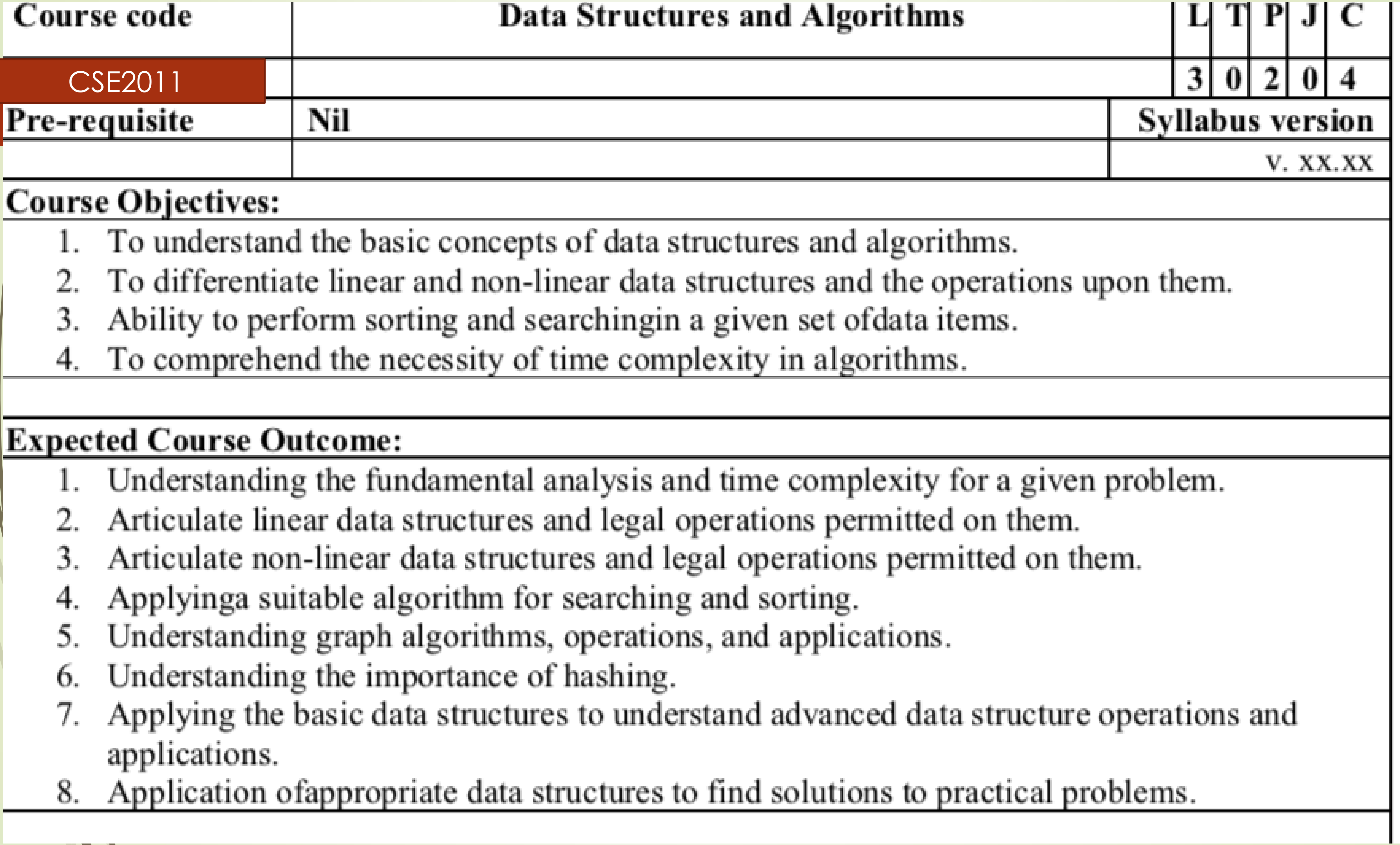
Question Pattern for Quizzes and CATs

- MCQs and short answers can be typed (by students) in MOODLE
- Diagrams / mathematical derivations / rough work / schemes / flow charts and similar other materials can be uploaded by the student for evaluation.
- MCQs
  - more than one correct answer model can be followed.
  - Questions and choices shall also be shuffled

Attendance

- student class attendance is mandatory.
- Student shall be permitted to appear for CAT – I and CAT – II only if he / she has a minimum of 75% attendance (as per rule 9.0 pointers are exempted).
- Answer sheets will not be evaluated even if he / she appears for the examination.
- Appearing for the examination, by debarred students, can be considered as a ‘practice’ for those students. This might help him / her to enhance the learning outcomes.
- However, mark posting will be disabled in VTOP for students who shall be debarred for CAT I and II.

CSE2011



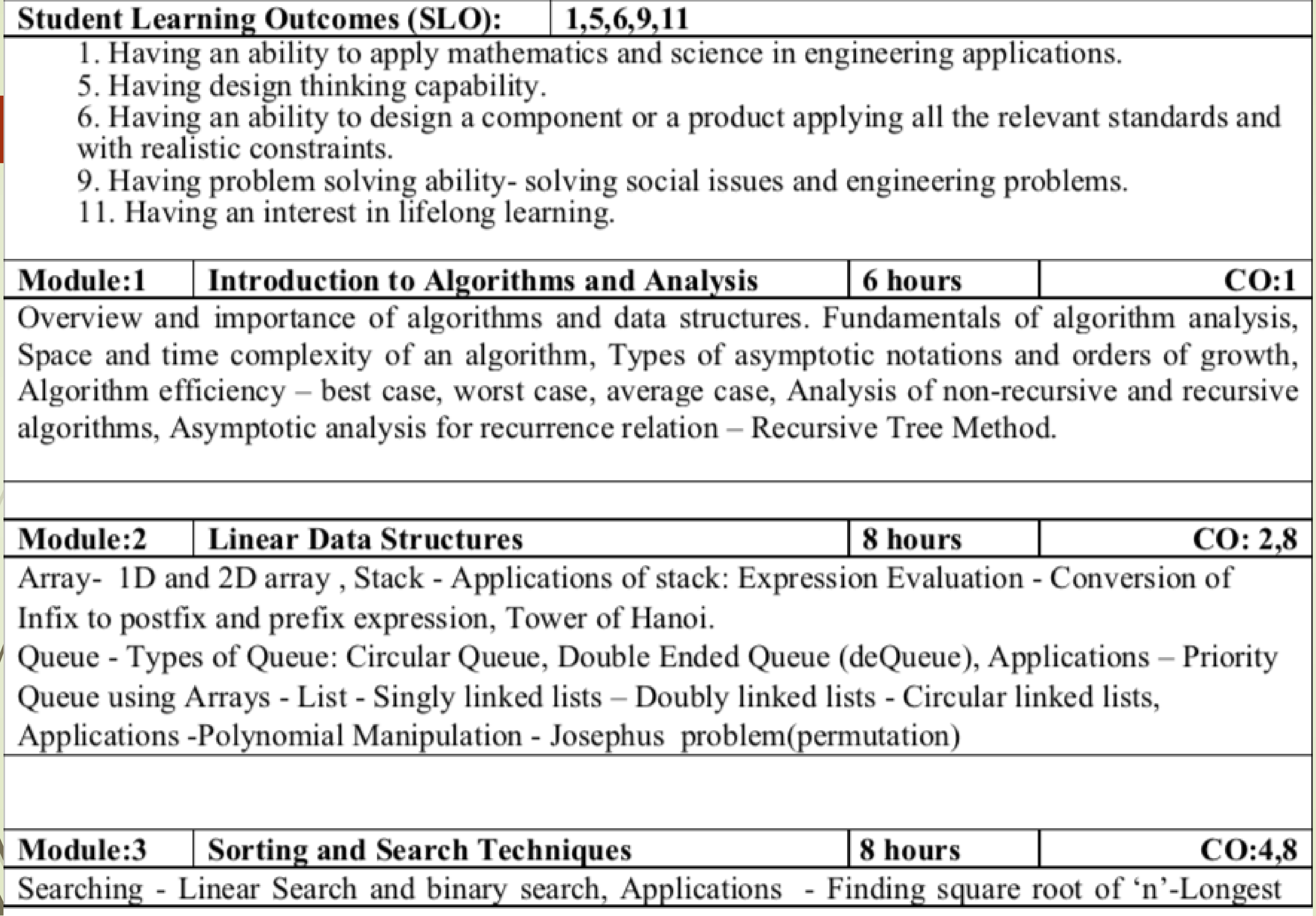
Course code	Data Structures and Algorithms	L	T	P	J	C
		3	0	2	0	4
Pre-requisite	Nil	Syllabus version				
		v. xx.xx				

Course Objectives:

1. To understand the basic concepts of data structures and algorithms.
2. To differentiate linear and non-linear data structures and the operations upon them.
3. Ability to perform sorting and searchingin a given set ofdata items.
4. To comprehend the necessity of time complexity in algorithms.

Expected Course Outcome:

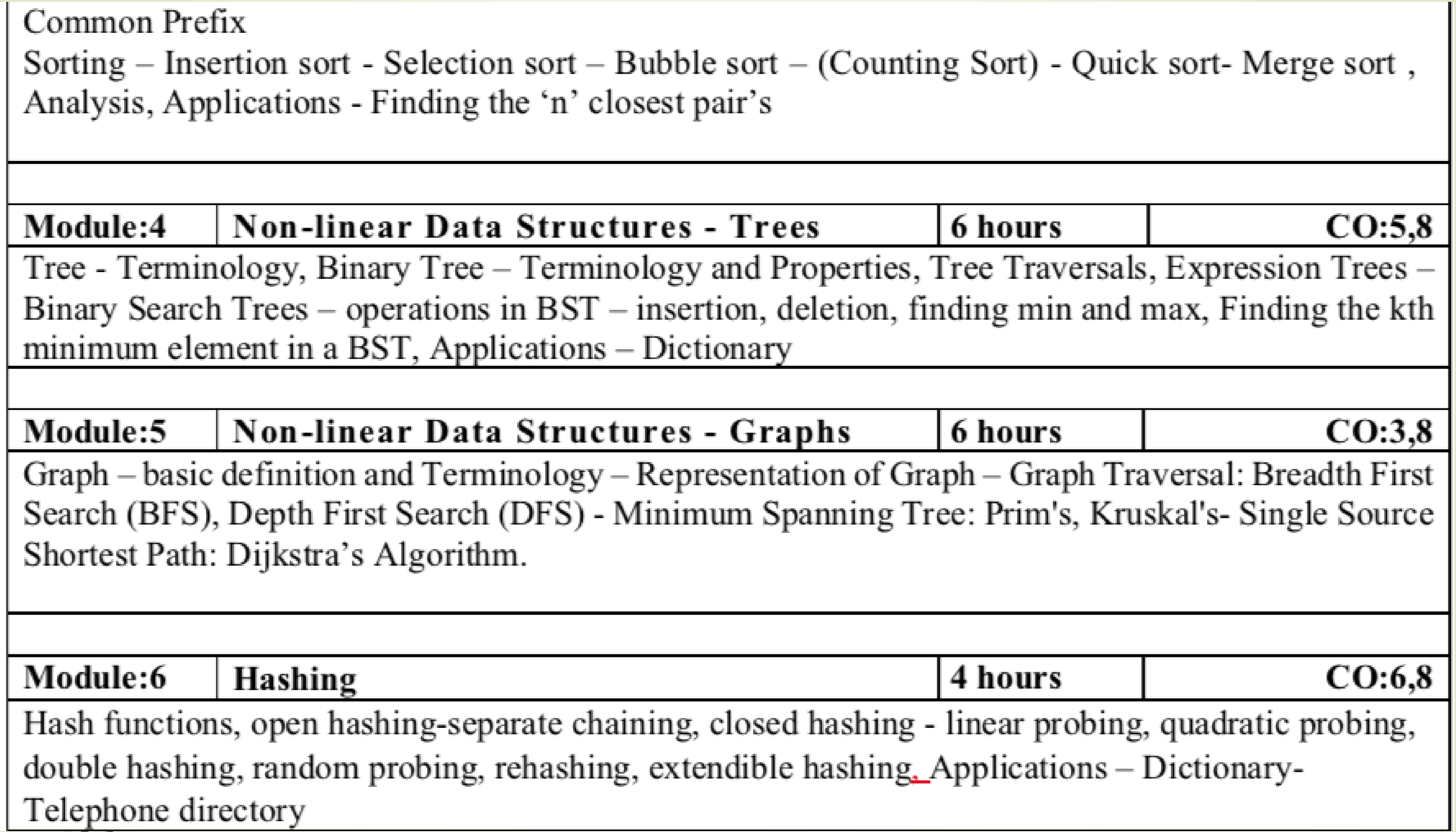
1. Understanding the fundamental analysis and time complexity for a given problem.
2. Articulate linear data structures and legal operations permitted on them.
3. Articulate non-linear data structures and legal operations permitted on them.
4. Applyinga suitable algorithm for searching and sorting.
5. Understanding graph algorithms, operations, and applications.
6. Understanding the importance of hashing.
7. Applying the basic data structures to understand advanced data structure operations and applications.
8. Application ofappropriate data structures to find solutions to practical problems.



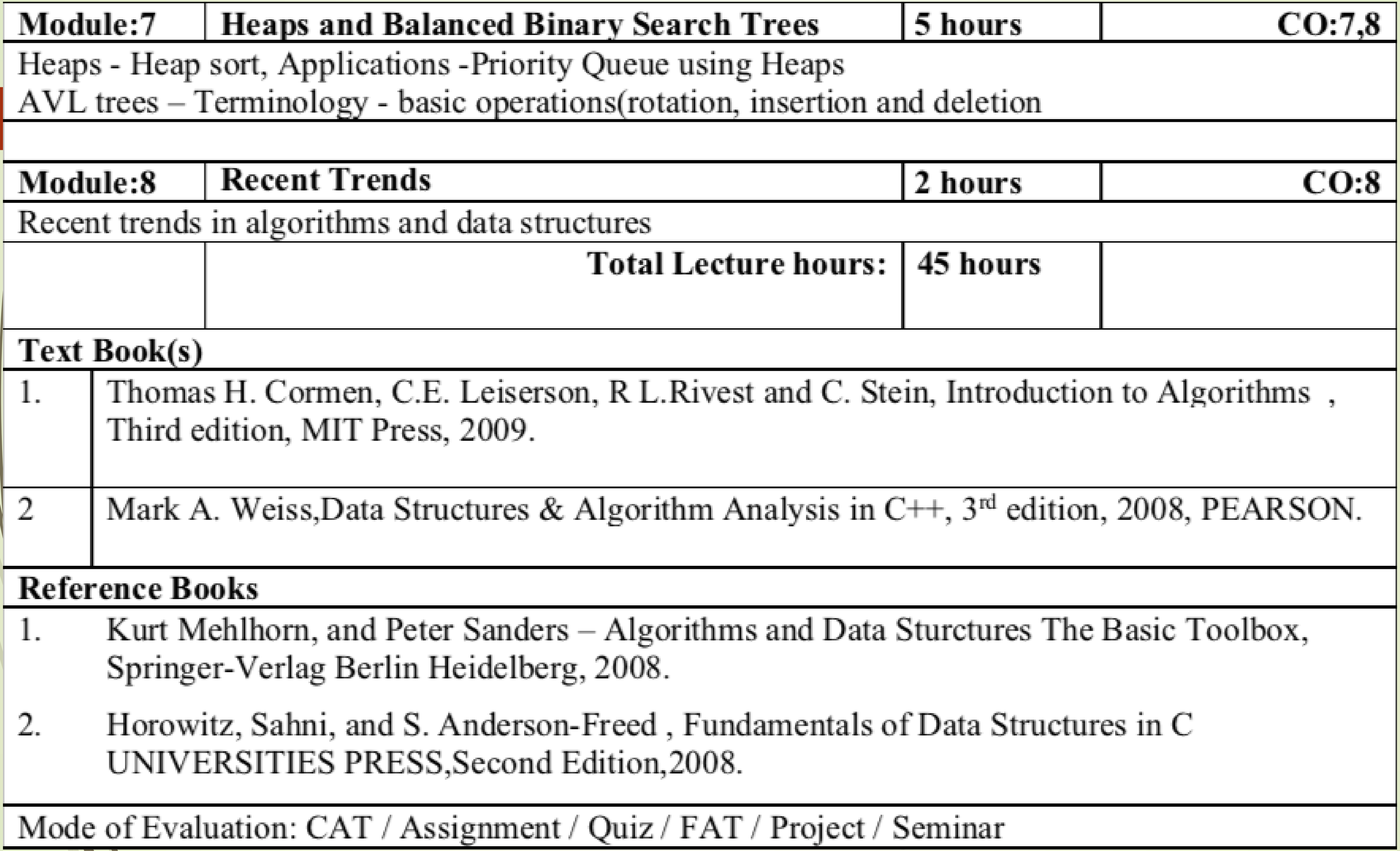
Student Learning Outcomes (SLO): 1,5,6,9,11

1. Having an ability to apply mathematics and science in engineering applications.
5. Having design thinking capability.
6. Having an ability to design a component or a product applying all the relevant standards and with realistic constraints.
9. Having problem solving ability- solving social issues and engineering problems.
11. Having an interest in lifelong learning.

Module:1	**Introduction to Algorithms and Analysis**	**6 hours**	**CO:1**
Overview and importance of algorithms and data structures. Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case, average case, Analysis of non-recursive and recursive algorithms, Asymptotic analysis for recurrence relation – Recursive Tree Method.			
Module:2	**Linear Data Structures**	**8 hours**	**CO: 2,8**
Array- 1D and 2D array , Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Tower of Hanoi. Queue - Types of Queue: Circular Queue, Double Ended Queue (deQueue), Applications – Priority Queue using Arrays - List - Singly linked lists – Doubly linked lists - Circular linked lists, Applications -Polynomial Manipulation - Josephus problem(permutation)			
Module:3	**Sorting and Search Techniques**	**8 hours**	**CO:4,8**
Searching - Linear Search and binary search, Applications - Finding square root of 'n'-Longest			



Common Prefix Sorting – Insertion sort - Selection sort – Bubble sort – (Counting Sort) - Quick sort- Merge sort , Analysis, Applications - Finding the ‘n’ closest pair’s			
Module:4	**Non-linear Data Structures - Trees**	**6 hours**	**CO:5,8**
Tree - Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, finding min and max, Finding the kth minimum element in a BST, Applications – Dictionary			
Module:5	**Non-linear Data Structures - Graphs**	**6 hours**	**CO:3,8**
Graph – basic definition and Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) - Minimum Spanning Tree: Prim's, Kruskal's- Single Source Shortest Path: Dijkstra’s Algorithm.			
Module:6	**Hashing**	**4 hours**	**CO:6,8**
Hash functions, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing, random probing, rehashing, extendible hashing, Applications – Dictionary-Telephone directory			



Module:7	**Heaps and Balanced Binary Search Trees**	**5 hours**	**CO:7,8**
Heaps - Heap sort, Applications -Priority Queue using Heaps AVL trees – Terminology - basic operations(rotation, insertion and deletion			
Module:8	**Recent Trends**	**2 hours**	**CO:8**
Recent trends in algorithms and data structures			
	Total Lecture hours:	**45 hours**	
Text Book(s)			
1.	Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms , Third edition, MIT Press, 2009.		
2	Mark A. Weiss,Data Structures & Algorithm Analysis in C++, 3rd edition, 2008, PEARSON.		
Reference Books			
1.	Kurt Mehlhorn, and Peter Sanders – Algorithms and Data Sturctures The Basic Toolbox, Springer-Verlag Berlin Heidelberg, 2008.		
2.	Horowitz, Sahni, and S. Anderson-Freed , Fundamentals of Data Structures in C UNIVERSITIES PRESS,Second Edition,2008.		
Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar			

Course objectives

- **Learn basic data structures and algorithms**
    - data structures – how data is organized
    - algorithms – unambiguous sequence of steps to compute something
    - algorithm analysis – determining how long an algorithm will take to solve a problem

- **Become a better software developer**
    - "Data Structures + Algorithms = Programs"

Algorithms

- Design
- Domain Knowledge
- Any language
- H/W and OS independent
- Analyze

Programs

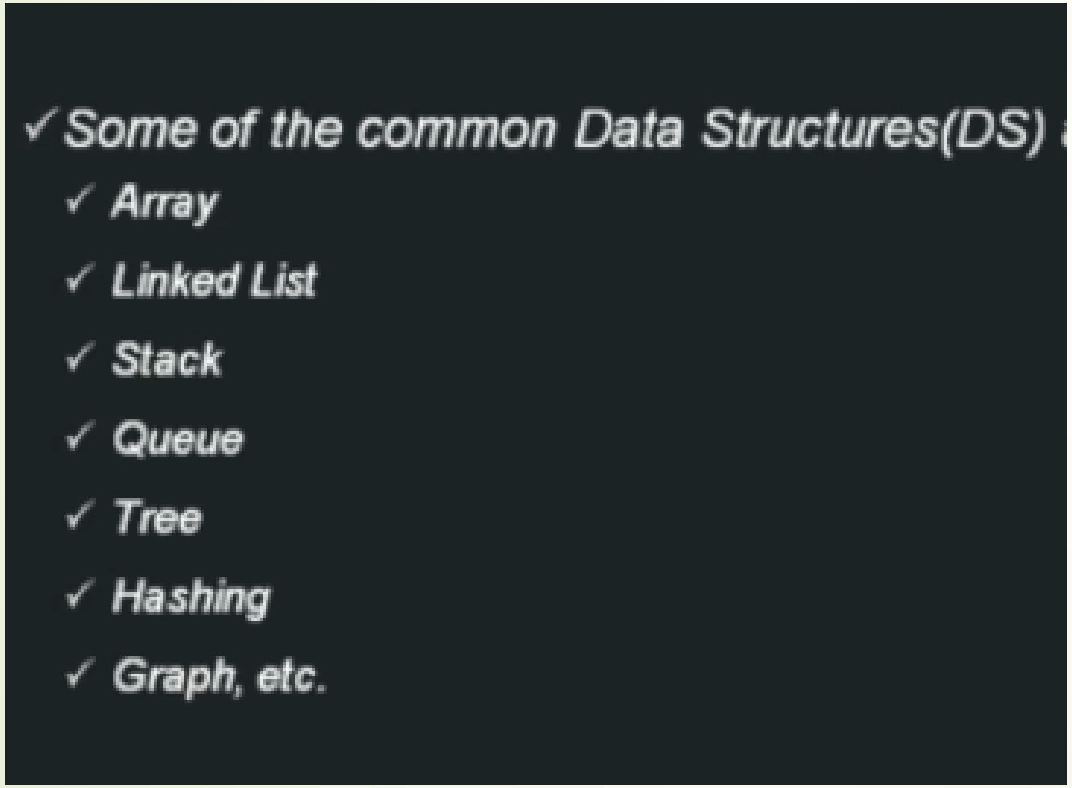
- Implementation
- Programmer
- Programming language
- H/W and OS dependent
- Testing

Session Objectives

- To understand the concepts of
  - Data structures
  - Types of Data Structures
  - Applications
  - Algorithms

What is Data Structure?

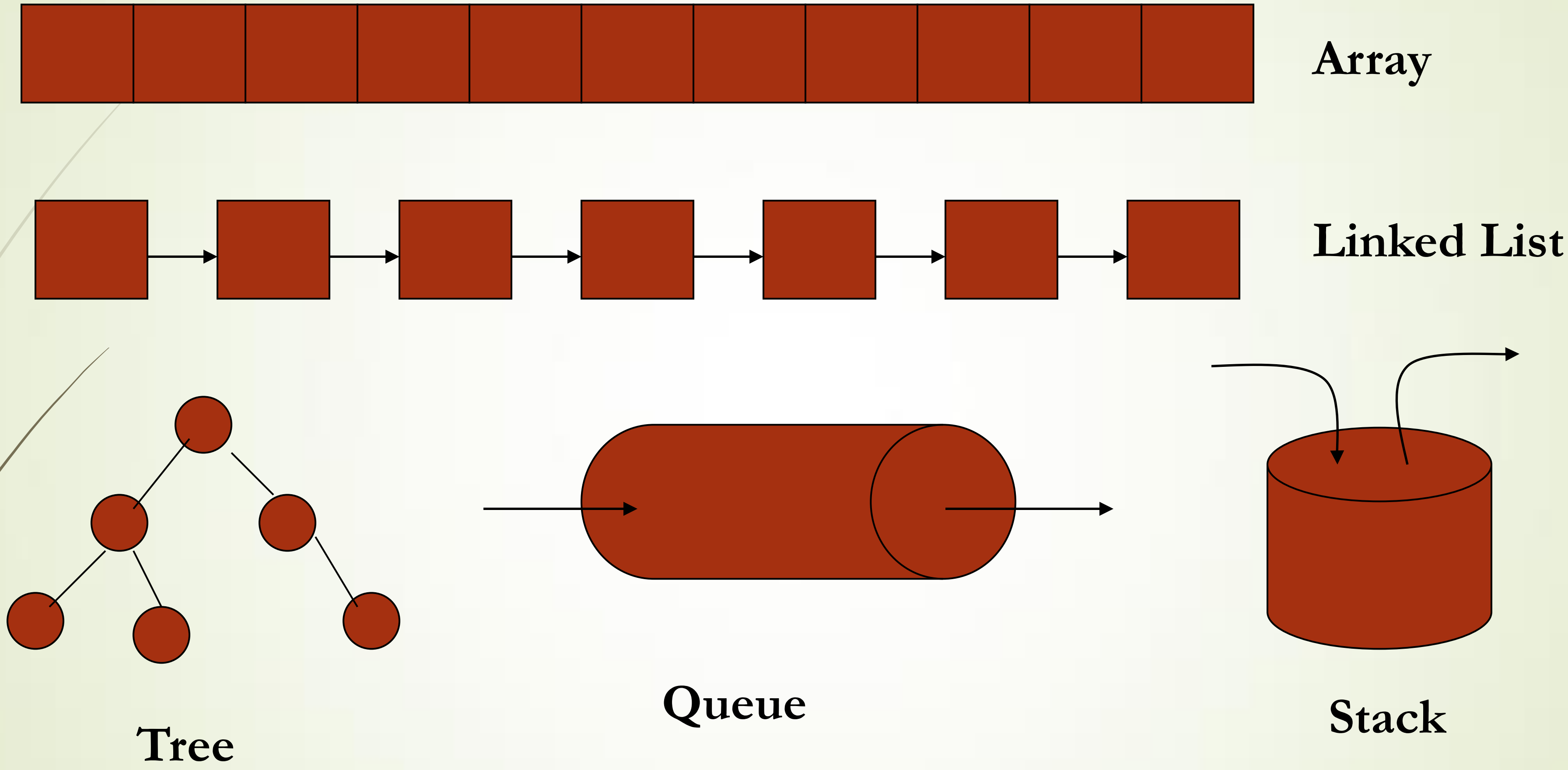
Data structure is a way to “organize data” that enables it to be processed in an efficient time



✓Some of the common Data Structures(DS)

- ✓ Array
- ✓ Linked List
- ✓ Stack
- ✓ Queue
- ✓ Tree
- ✓ Hashing
- ✓ Graph, etc.

Types of data structures

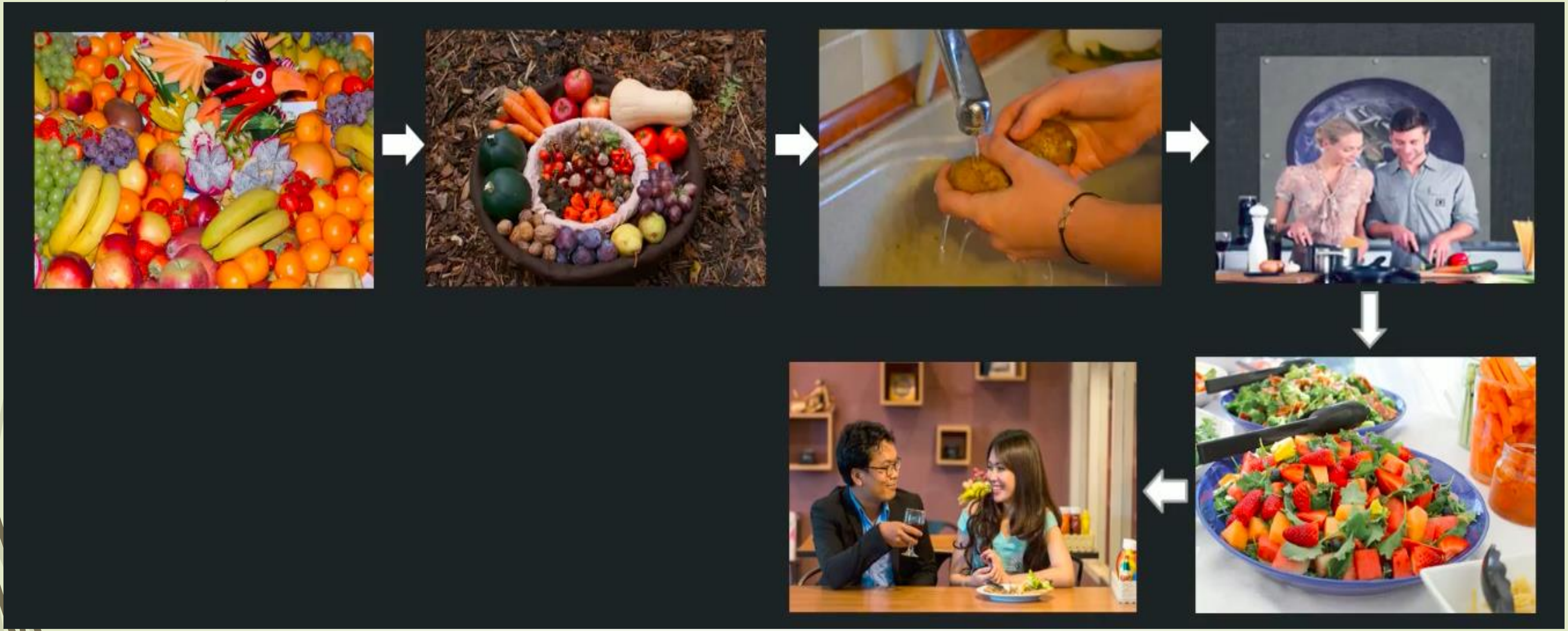


There are many, but we named a few. We’ll learn these data structures in great detail!

What is an Algorithm?

A set of rules to be followed to solve a problem

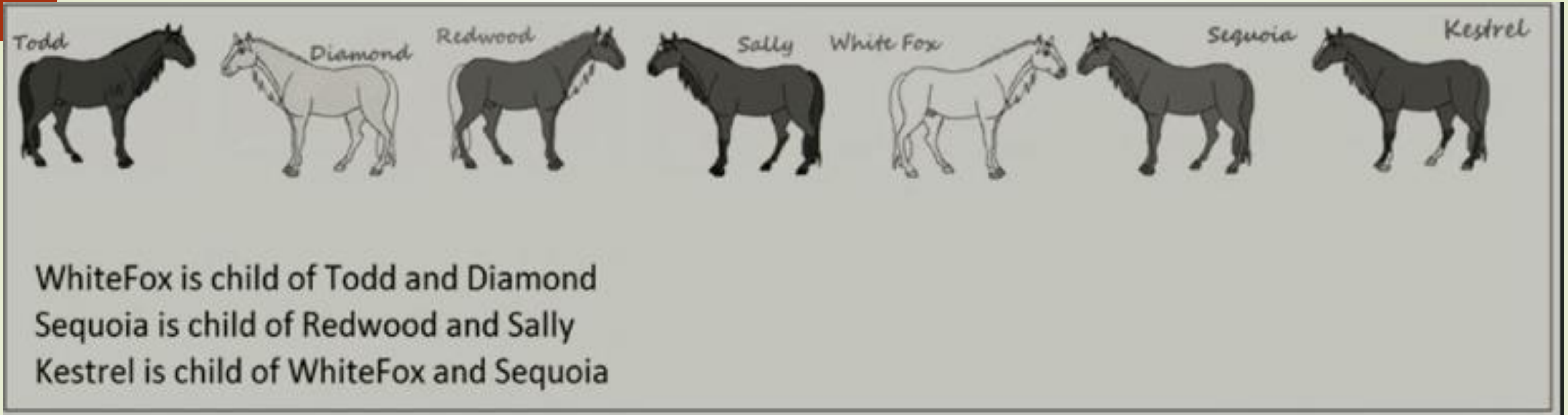
Example- Lets say we want to prepare a homemade salad for dinner



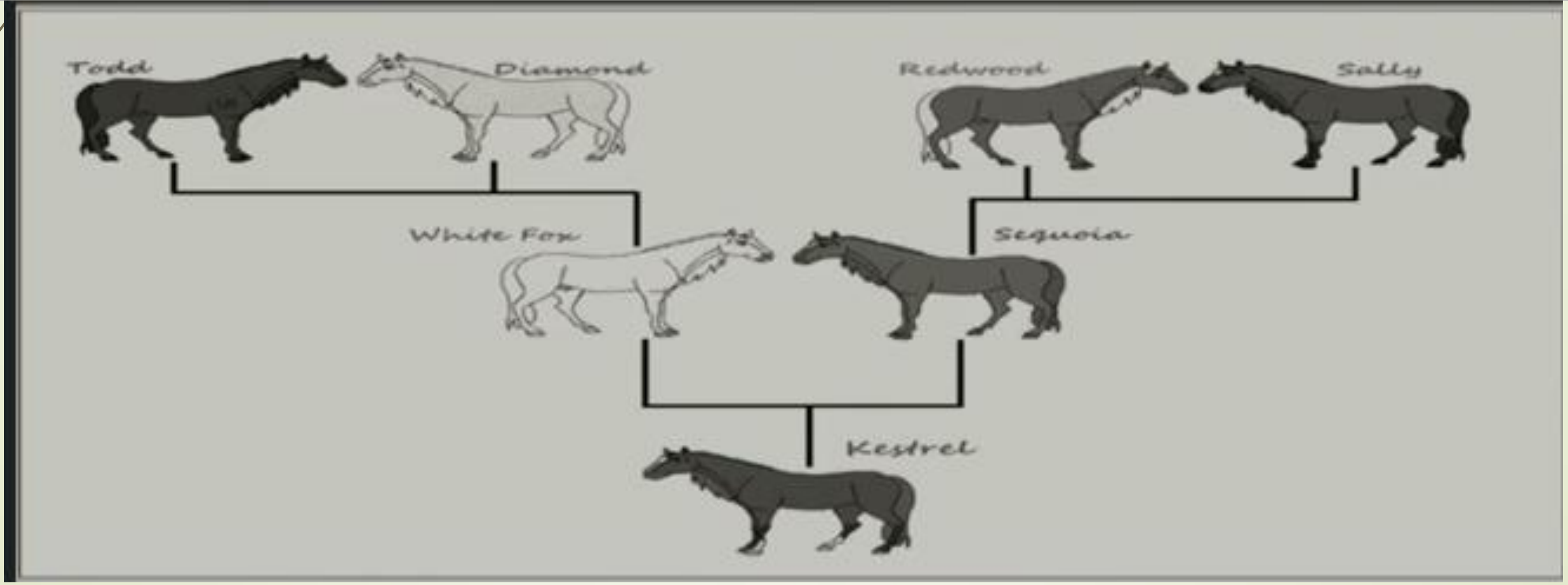
Real life examples

Queue

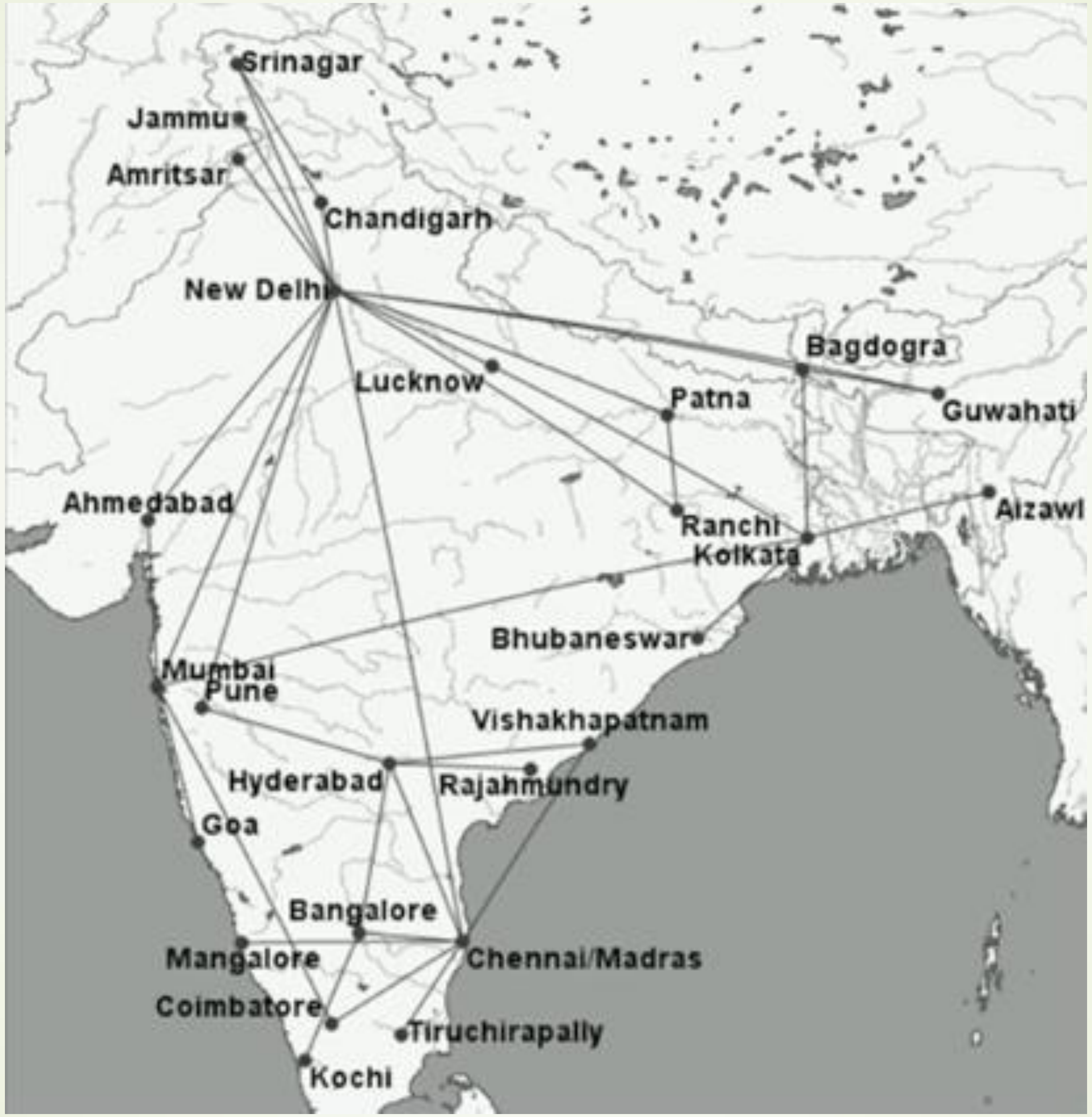
Tree

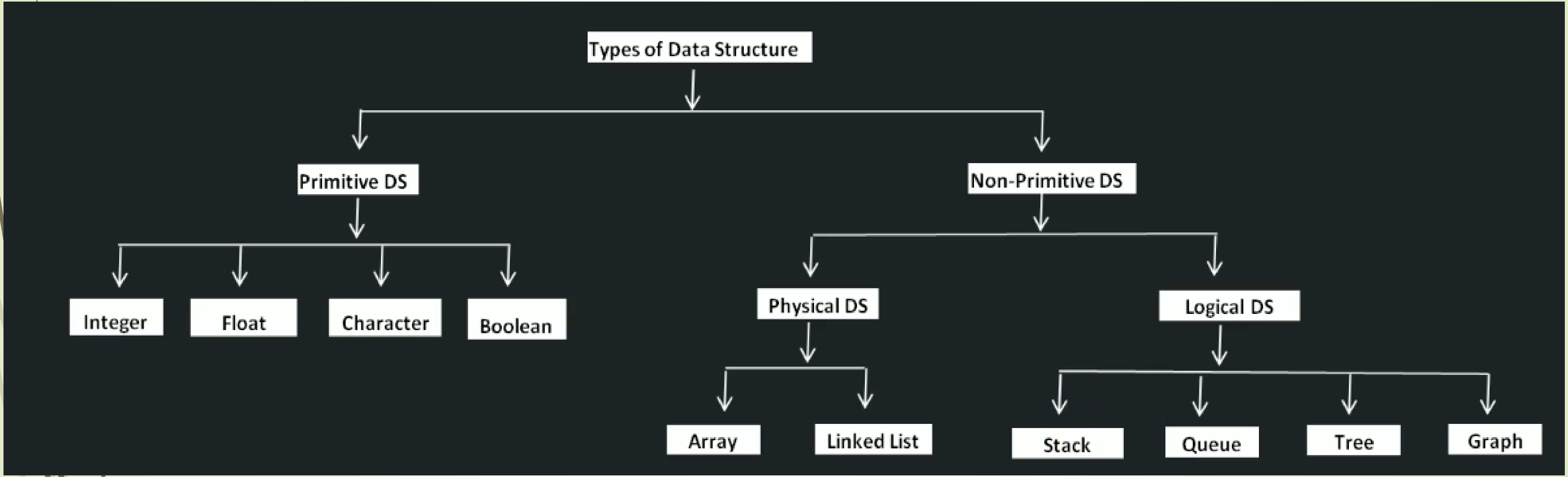


WhiteFox is child of Todd and Diamond
Sequoia is child of Redwood and Sally
Kestrel is child of WhiteFox and Sequoia



Graphs


Srinagar
Jammu
Amritsar
Chandigarh
New Delhi
Lucknow
Bagdogra
Patna
Guwahati
Ahmedabad
Ranchi
Kolkata
Aizawl
Bhubaneswar
Mumbai
Pune
Vishakhapatnam
Hyderabad
Rajahmundry
Goa
Bangalore
Mangalore
Chennai/Madras
Coimbatore
Tiruchirapally
Kochi



Searching
Sorting
Insertion
Deletion
Updation

The data structures can also be classified on the basis of the following characteristics:

Characterstic	Description
Linear	In Linear data structures,the data items are arranged in a linear sequence. Example: **Array**
Non-Linear	In Non-Linear data structures,the data items are not in sequence. Example: **Tree**, **Graph**
Homogeneous	In homogeneous data structures,all the elements are of same type. Example: **Array**
Non-Homogeneous	In Non-Homogeneous data structure, the elements may or may not be of the same type. Example: **Structures**
Static	Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: **Array**
Dynamic	Dynamic structures are those which expands or shrinks depending upon the program need and its execution. Also, their associated memory locations changes. Example: **Linked List created using pointers**

Thank you