Class- XII

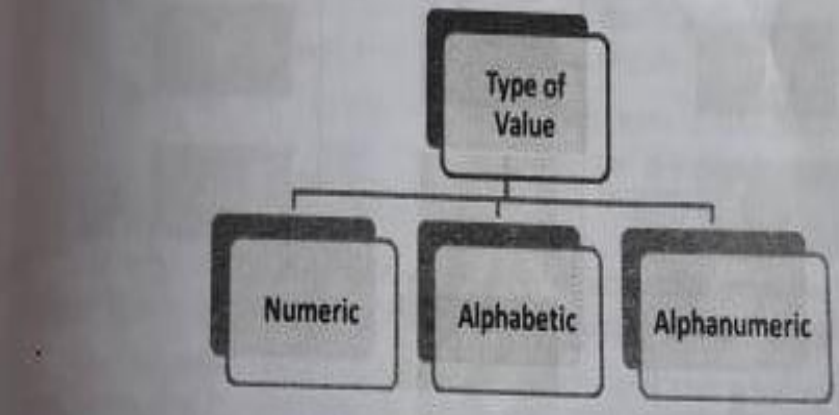
Subject- IT & Programming

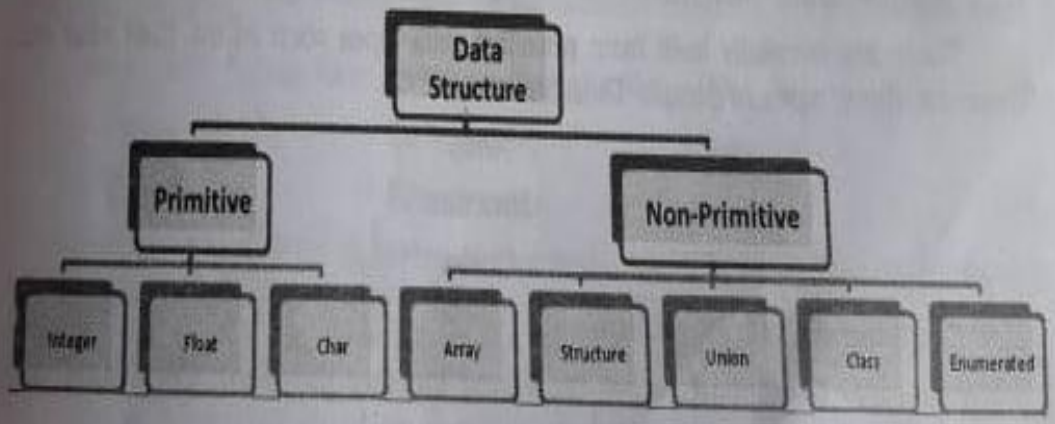
Ch.-1 Data Structure

DATA STRUCTURE : AN INTRODUCTION

1.1 DATA

Any fact, guess, observation, occurrence of event etc., which is meaningless to user and used to input to computer is celled Data. Data can be classified on two basis: (i) Type of value (ii) Data Structure.





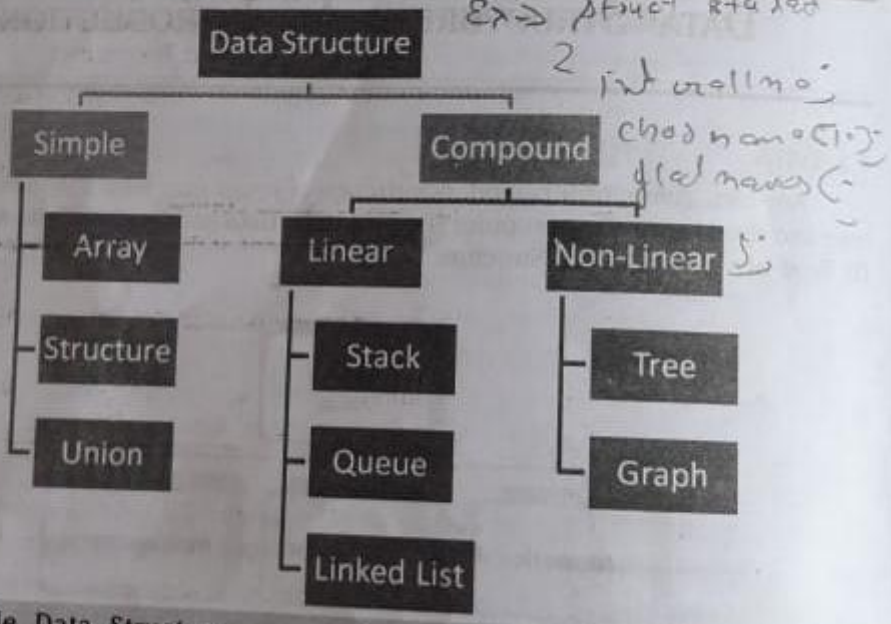
Primitive : Also called Fundamental data type, these are those data types which are not composed of other data type.e.g.: Integer, floating point, character.

Non Primitive: Also called Derived data type or User defined data type, these data types are composed of primitive data types. E.g. - Enumerated, Array, Structure etc.

1.2 DATA STRUCTURE

- Before the data is being processed, it has to be organised. Organising data in a required manner is called **data structure**.
- *"A Data structure is a named collection of different types of data which can be processed together as a single unit."*
- Data structure is very important in programming because they not only allow the user to combine various data types in a group but also allow processing of group as a single unit, thereby making the processing easier & faster.

1.3 CLASSIFICATION OF DATA STRUCTURE



1.3.1 Simple Data Structure

There are normally built from primitive data types such as int, float char etc. There are three types of Simple Data Structures :

Array
Structure
Union

1.3.2 Compound Data Structure

When Simple data types are combined together they form Compound data structure. They are further classified as-

(i) Linear : These data structure are single level (layer) Structure. In other words a data structure is said to be linear if its element form a sequence. For example-

- Stack
- Queue
- Linked List

(ii) Non - Linear : These are multilevel (layer) data structure.Their elements do not follow any sequence For Example -

- Tree
- Graph

1.4 OPERATION IN DATA STRUCTURE

In computer, when data are organized using any of the data structure then following basic operations can be performed on them ~

S.No.	Operation	Description of Operation
1.	Traversing	The task of accessing and processing each and every element/ data in the data structure is called Traversing.
2.	Searching	The task of finding position of an element in the data structure is called Searching.
3.	Sorting	The task of arranging elements of the data structure in ascending or descending order is called Sorting.
4.	Insertion	The task of adding an element at the given position in the data structure is called Insertion.
5.	Deletion	The task of removing an element of the given position from the data structure is called Deletion.
6.	Merging	The task of combining elements of two similar data structure to form a new data structure is called Merging.

1.5 ALGORITHM

An algorithm can be defined as a set of sequential steps, written in ordinary English language (Human understandable language) to solve a problem.

An algorithm has no standard syntax there by it varies from person to person. Also, it may be possible that there are more than one correct algorithm for the same problem.

An algorithm must have following characterstics -

- Finiteness
- Definiteness
- Input
- Output
- Effectiveness

The efficiency of an algorithm is measured in terms of **Time complexity** and **Space complexity.**

1.5.1 Time Complexity

The time complexity of an algorithm is the amount of computer time that it needs to run to completion. The exact time we would determine would not apply to many machines or to any machine. This is called **"Time complexity".**

Since the efficiency of the algorithm vary from machine to machine and different with different input data, we consider the worst case of execution to measure time complexity.

The Big - O Notation is normally used to represent the computing time of algorithm. When we say that the computing time of an algorithm is O(g(n)), we means that its execution takes no more than a constant time g(n). Here n is a parameter which characterises the no. of inputs or outputs.

The order of some of the commonly used algorithm are

Algorithm	Order
1. sequential search	O(n)
2. binary search	$O(\log_2 n)$
3. selection sort,bubble sort	$O(n^2)$
4. quick sort,healp sort	$O(n \log_2 n)$
5. matrix addition,	$O(n^2)$
6. matrix multiplication	$O(n^3)$

1.5.2 Space Complexity

Space complexity of an algorithm is the memory space occupied by it during execution. The space needed by a program is the sum of two components :

(a) Fixed space requirement

(b) Variable space requirement

Fixed space : It includes the instruction space, space for simple variable, fixed sized structured variables and constant.

Variable space : It includes space needed by structured variable whose space depends on a particular instance, space required for recursive function.

QUESTIONS

[A] Multiple Choice Question

Q.1 In determining efficiency of algorithm the space complexity is measured in terms of :

(a) Maximum memory required by the algorithm

(b) minimum memory required by the algorithm

(c) Average memory required by the algorithm

(d) Maximum disk space required by the algorithm

Q.2 In determining efficiency of algorithm the time complexity is measured in terms of :

(a) Counting the microseconds

(b) Counting the number of operations

(c) Counting size(in kilobyte) of algorithm

(d) None of these