INDRAPRASTHA NEW ARTS COMMERCE AND SCIENCE COLLEGE WARDHA

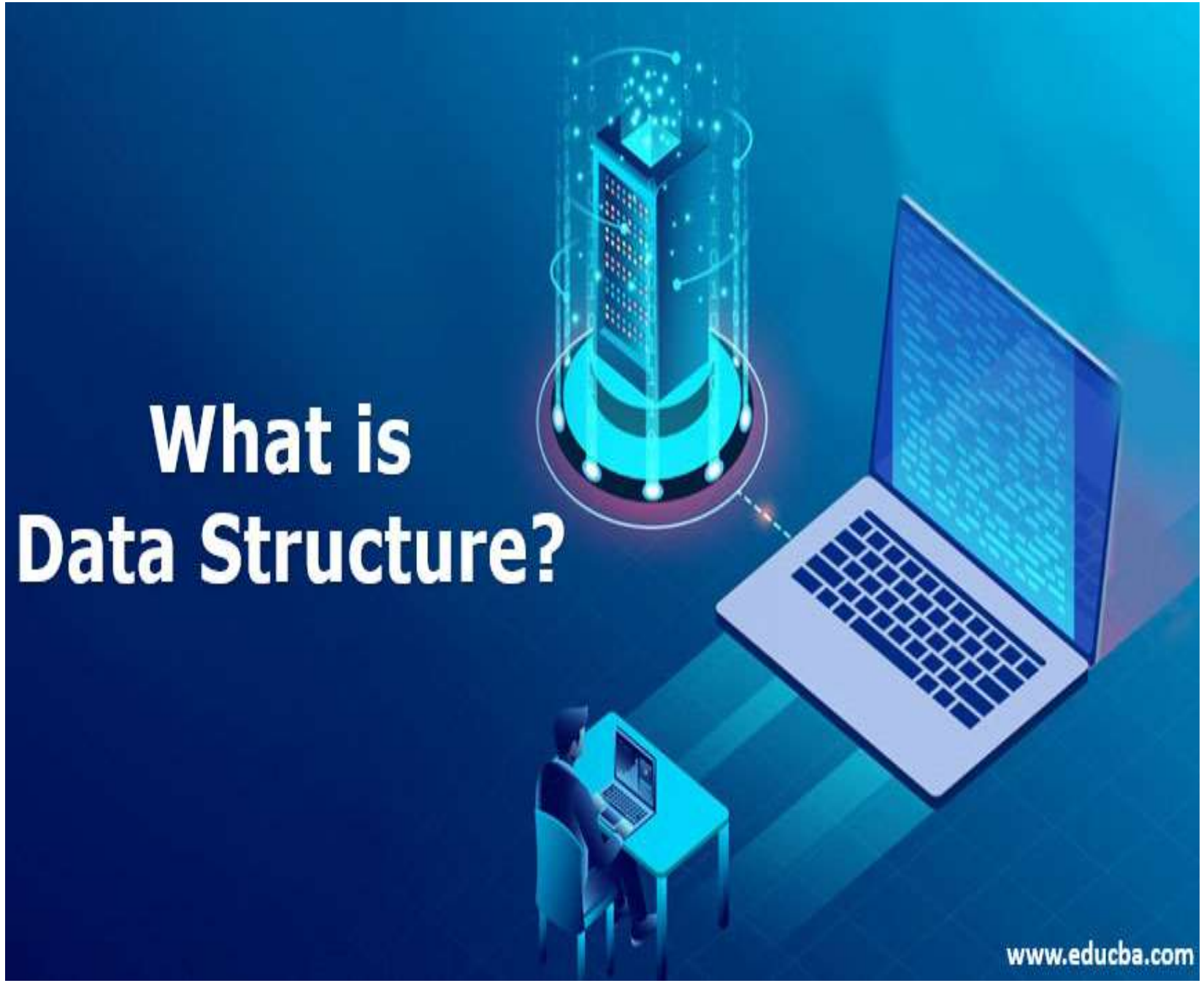
DATA STRUCTURE
BSC III SEM

Prof.Bharti Yerane

bhartiyerne04997@gmail.com

Introduction to Data Structure

- Computer is an electronic machine which is used for data processing and manipulation.
- When programmer collects such type of data for processing, he would require to store all of them in computers main memory.
- In order to make how computer work we need to know
  - Representation of data in computer.
  - Accessing of data.
  - How to solve problem step by step.
- For doing all of this task we used Data Structure



What is Data Structure

- A data structure is a specialized format for organizing, processing, retrieving and storing data.
- In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms.

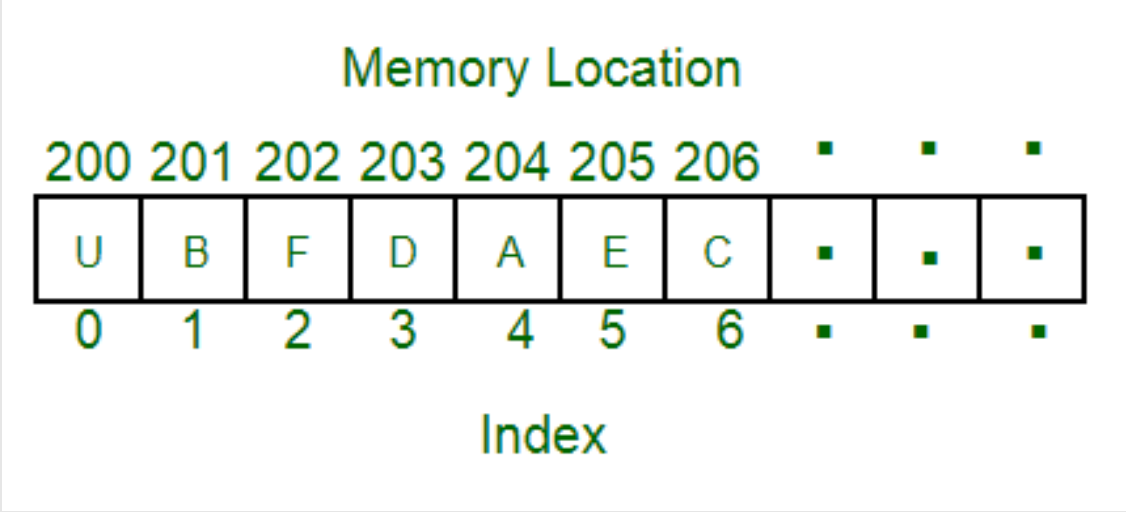
What is Data Structure

- Data Structure can be defined as the group of data elements which provides an efficient way of storing and organizing data in the computer so that it can be used efficiently.
- examples of Data Structures are arrays, Linked List, Stack, Queue, etc.
- Data Structures are widely used in almost every aspect of Computer Science i.e. Operating System, Compiler Design, Artificial intelligence, Graphics and many more.
- Data Structures are the main part of many computer science algorithms as they enable the programmers to handle the data in an efficient way.
- It plays a vital role in enhancing the performance of a software or a program as the main function of the software is to store and retrieve the user's data as fast as possible

Data Structure

- A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.

- For example, we can store a list of items having the same data-type using the *array* data structure.

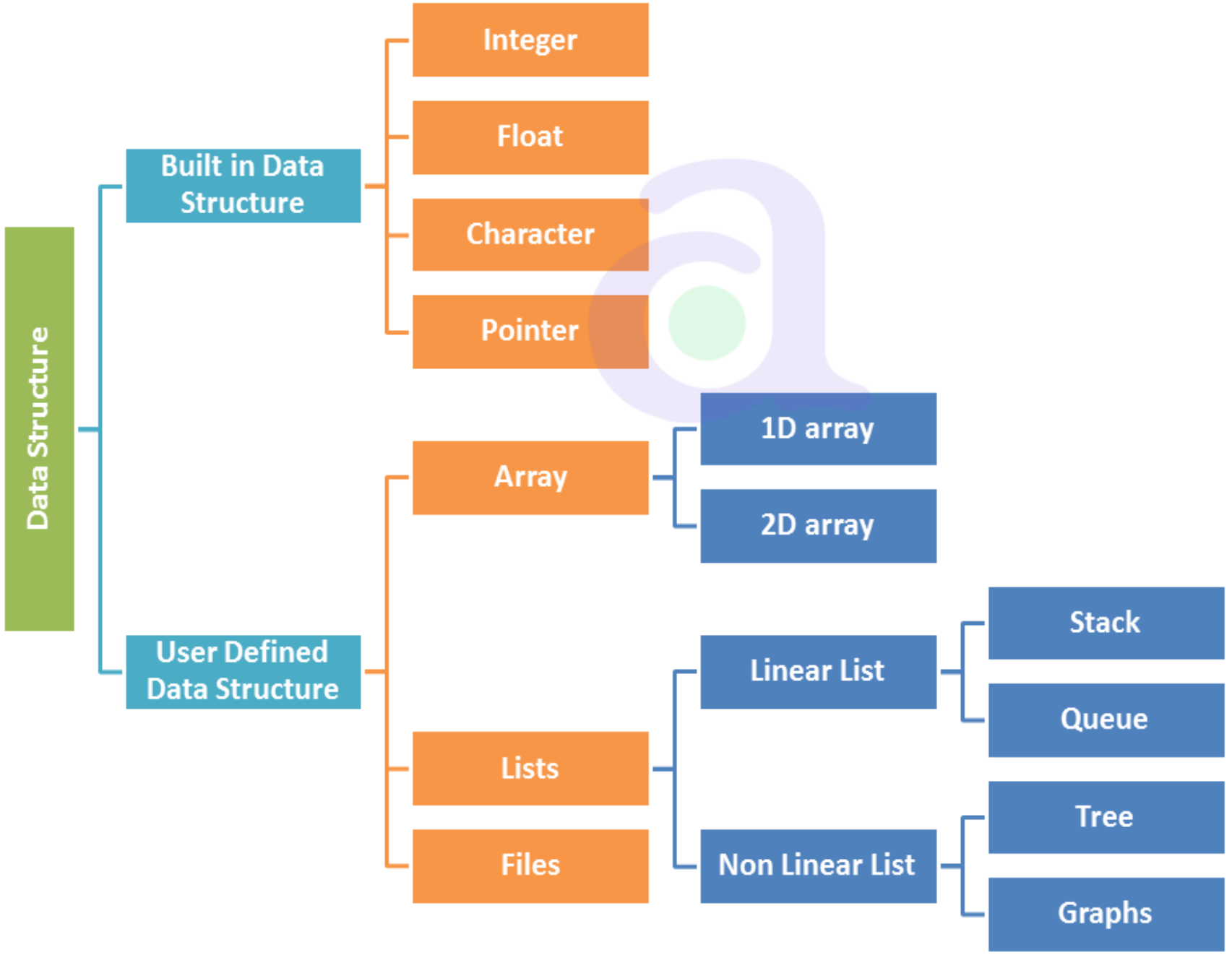


- The representation of particular data structure in the main memory of a computer is called as storage structure.
- The storage structure representation in auxiliary memory is called as file structure.
- It is define as the way of storing and manipulating data in organized form so that it can be used efficiently
- **Data Structure mainly specifies the following four things**:

  1)organization of data 2)accessing method 3)degree of associativity 4) processing alternative for information

- **Algorithm + Data Structure = Program**
- **Data Structure study Covers the following points**

  1) Amount of memory require to store
  2) Amount of time require to process
  3) Representation of data in memory
  4) Operations performs on data



Types Of DS

The DS are divided into two types:

1) Primitive
2) Non primitive

Non primitive divided into two type

1) Linear DS
2) Non linear DS

DATA TYPES

A particular kind of data item, as defined by the values it can take, the Programming language used, or the operations that can be performed on it.

- **Primitive Data Structure**

- Primitive Data Structure are basic structure and directly operated upon by machine instructions.
- Primitive data structures have different representations on different computers.
- Integers, floats, character and pointers are example of primitive data structures.
- These data types are available in most programming languages as built in type.

Integer: It is a data type which allows all values without fraction part. We can used it for whole numbers.

Float: It is a data type which is use for storing fraction numbers.

Character: It is a data type which is used for character values.

Pointer: A variable that hold memory address of another variable are called pointer.

Non Primitive Data Type

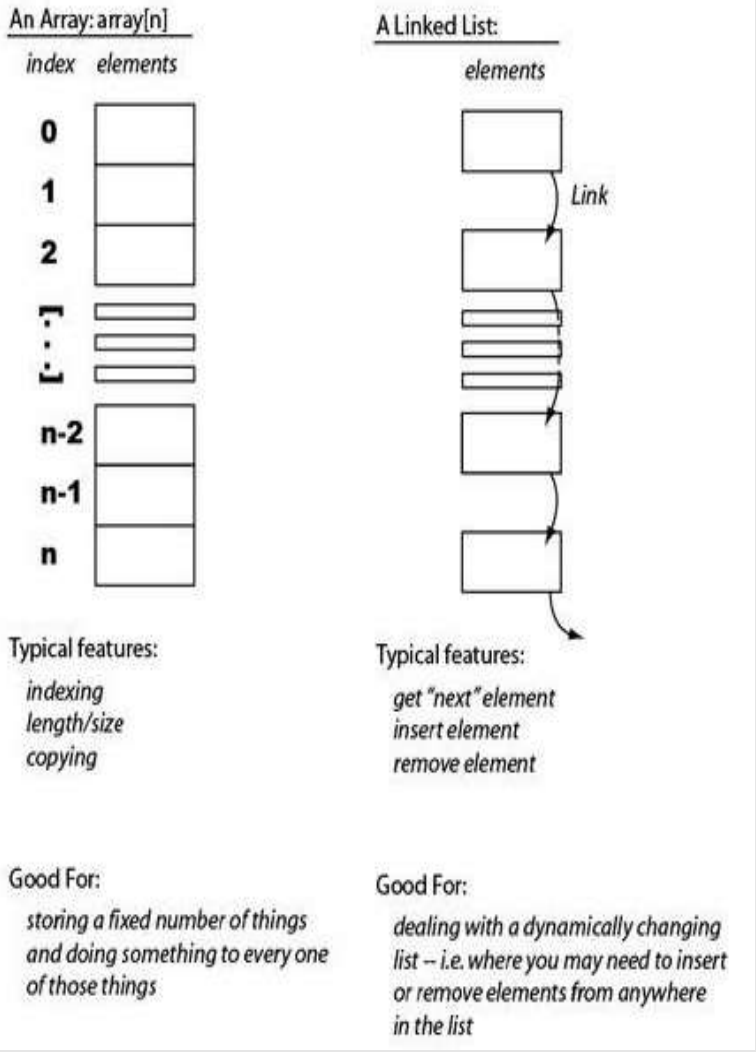
- These are more sophisticated data structures.
- These are derived from primitive data structure.
- The non – primitive data structures emphasize structuring of a group of homogeneous or heterogeneous data items.
- Example of non – primitive data types are Array, List, and File etc.
- A non – primitive data type is further divided into Linear and non – Linear data structure.

Array: An array is a fixed size sequenced collection of elements of the same data type.

List: An ordered set containing variable number of elements is called as List.

File: A file is a collection of logically related information. It can be viewed as a large list of records consisting of various fields.

Linear Data Structures



- A linear data structure simply means that it is a storage format of the data in the memory in which the data are arranged in contiguous blocks of memory.
- Example is the array of characters it represented by one character after another.
- In the linear data structure, member elements form a sequence in the storage.
- There are two ways to represent a linear data structure in memory.

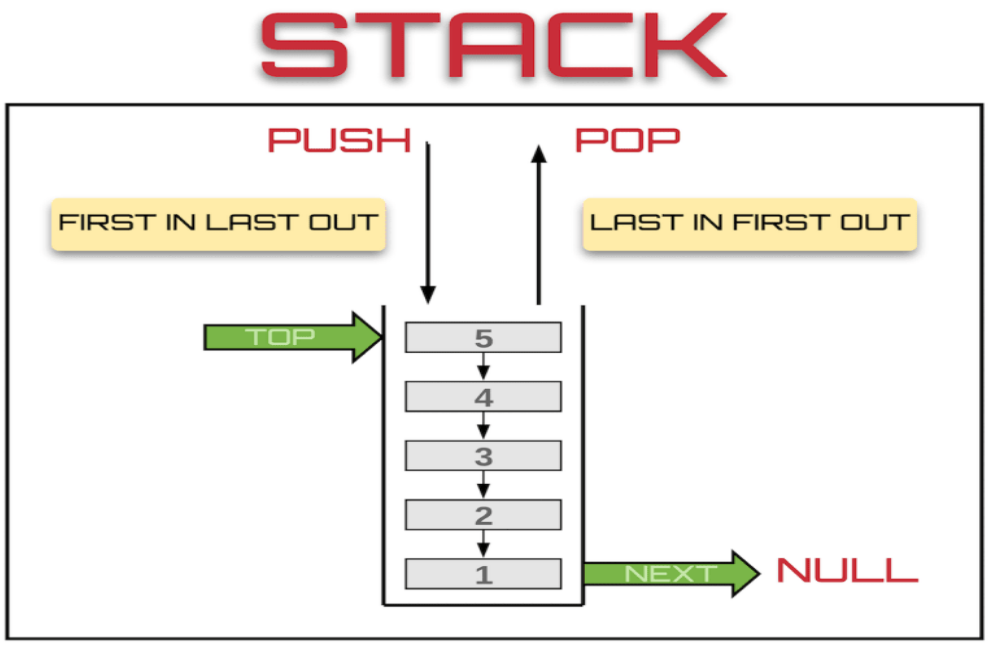
  static memory allocation

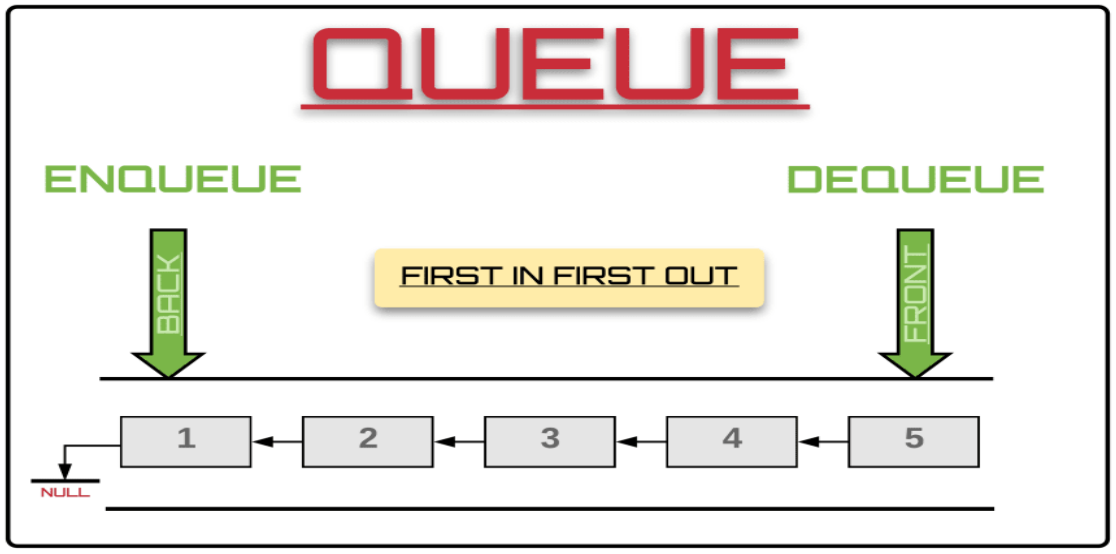
  dynamic memory allocation

The possible operations on the linear data structure are:

1) Traversing 2) Insertion 3) Deletion 4) searching 5) sorting

6) merging





- Example of Linear data structure are Stack and Queue

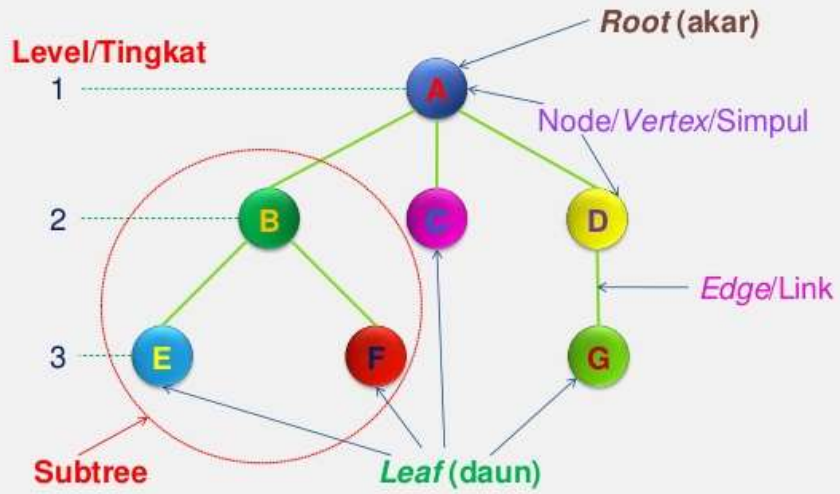
Stack

- Stack is a data structure in which insertion and deletion operations are performed at one end only.
- The insertion operation is referred to as 'PUSH' and deletion is referred as 'POP' operation
- Stack is also called as Last In First Out (LIFO) data structure.

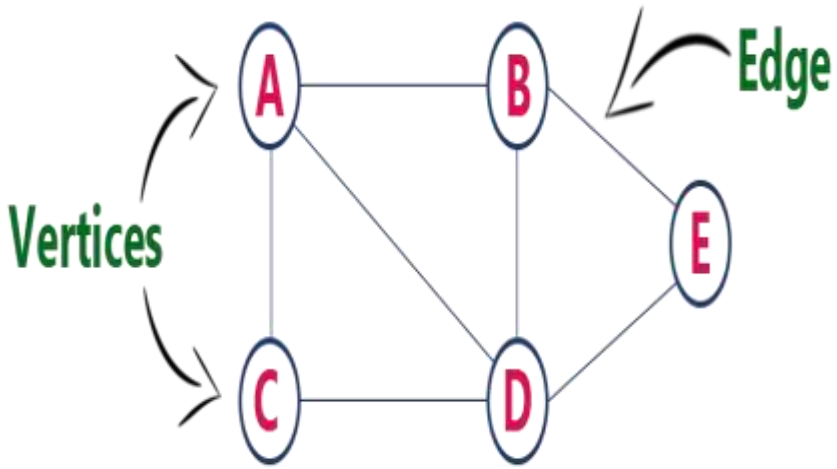
Queue

- The data structure which permits the insertion at one and deletion at another end, known as Queue.
- End at which deletion is occurs is known as FRONT end and another end at which insertion occurs is known as REAR end.
- Queue is also called as First In First Out (FIFO)

Components of Tree



Components of Graph



Non-Linear Data Structure

- Non linear DS are those data structure in which data items are not arranged in a sequence.
- Example on Non Linear DS are Tree and Graph.

TREE

- A Tree can be define as finite data items (nodes) in which data items are arranged in branches and sub branches
- Tree represent the hierarchical relationship between various elements
- Tree consist of nodes connected by edge, the represented by circle and edge lives connecting to circle.

Graph

- Graph is collection of nodes (information) and connecting edges (Logical relation) between nodes.
- A tree can be viewed as restricted graph
- Graph have many types: 1) Simple graph 2) Mixed graph 3) Multi graph 4) Directed graph 5) Un-directed graph

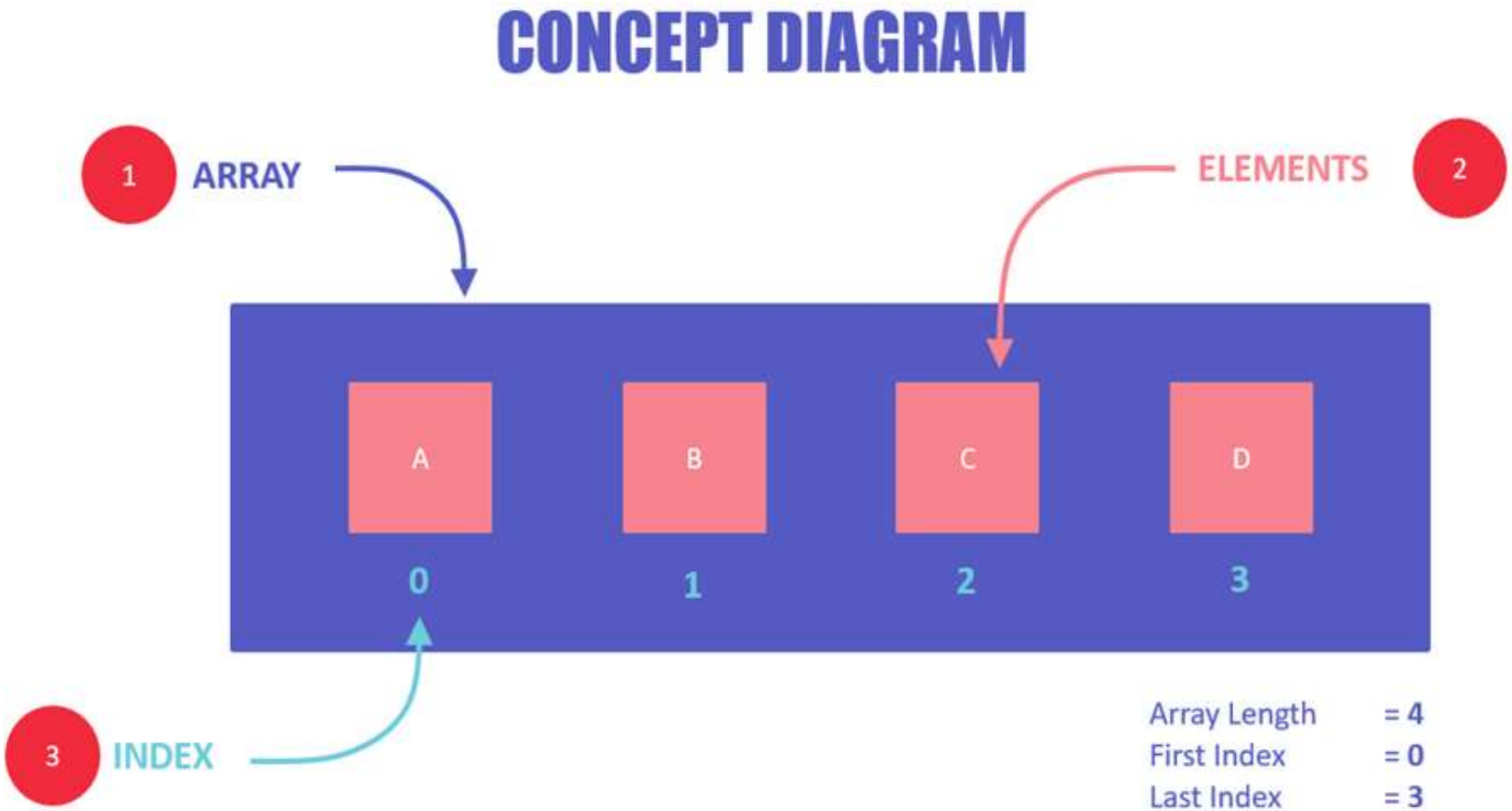
Difference Between Linear and Non Linear Data Structure

Linear Data Structure	Non – Linear Data Structure
◦ Every item is related to its previous and next item.	◦ Every item is attached with many other items.
◦ Data is arranged in linear sequence.	◦ Data is not arranged in sequence.
◦ Data items can be traversed in a single run	◦ Data cannot be traversed in a single run.
◦ E.g. Array, Stacks, Linked list, Queue	◦ E.g. Tree, Graph
◦ Implementation is easy.	◦ Implementation is difficult.

Operation on Data Structures

Design of efficient data structure must take operations to be performed on the DS into account. The most commonly used operations on DS are broadly categorized into following types

1. **Create:** This operation results in reserving memory for program elements. This can be done by declaration statement Creation of DS may take place either during compile-time or run-time.
2. **Destroy:** This operation destroy memory space allocated for specified data structure .
3. **Selection:** This operation deals with accessing a particular data within a data structure.
4. **Updation:** It updates or modifies the data in the data structure.
5. **Searching:** It finds the presence of desired data item in the list of data items, it may also find locations of all elements that satisfy certain conditions.
6. **Sorting:** This is a process of arranging all data items in a DS in particular order, for example either ascending order or in descending order.
7. **Splitting:** It is a process of partitioning single list to multiple list.
8. **Merging:** It is a process of combining data items of two different sorted list into single sorted list.
9. **Traversing:** It is a process of visiting each and every node of a list in systematic manner.



1. An array is a container of elements.
2. Elements have a specific value and data type, like "ABC", TRUE or FALSE, etc.
3. Each element also has its own index, which is used to access the element.

What are Arrays?

Array is a container which can hold a fix number of items and these items should be of the same type.

Most of the data structures make use of arrays to implement their algorithms.

•Following are the important terms to understand the concept of Array.

Element − Each item stored in an array is called an element.

Index − Each location of an element in an array has a numerical index, which is used to identify the element.

UNDERSTAND SYNTAX

PYTHON

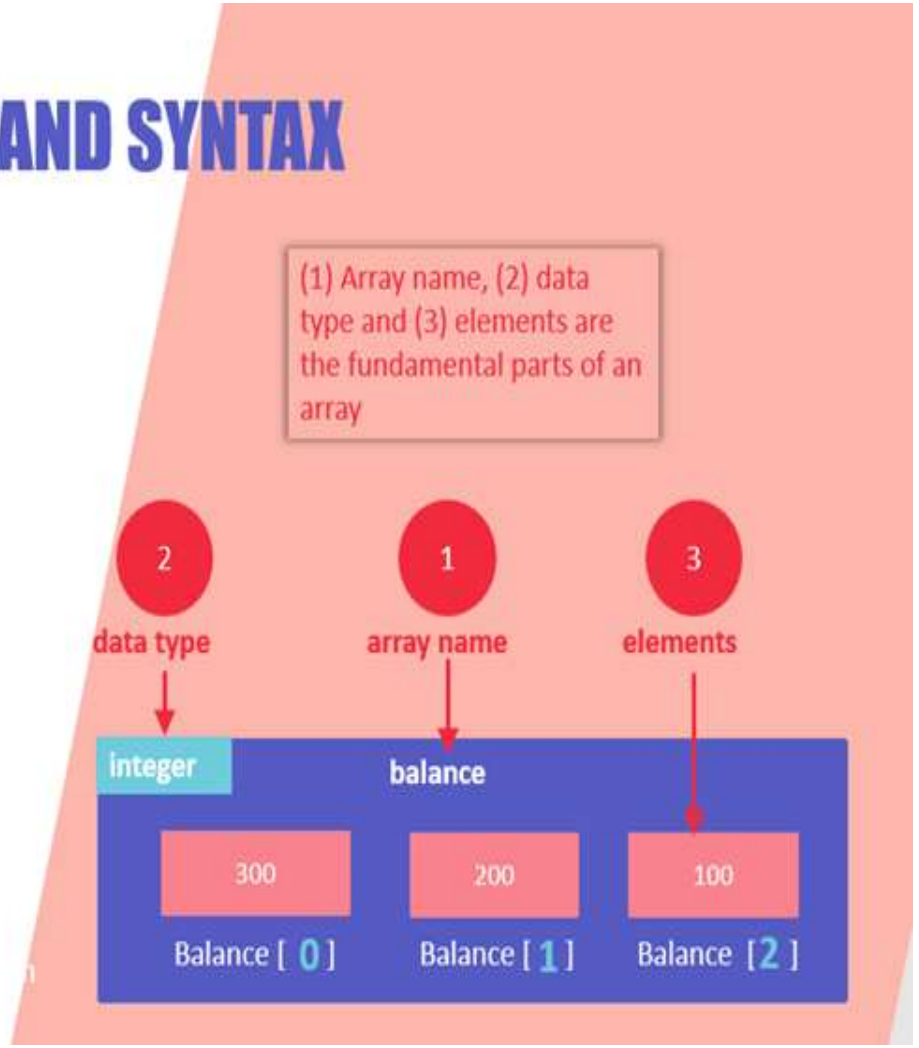
```
balance = array.array( 'i', [300,200,100] )
```

array name — data type — elements

C++

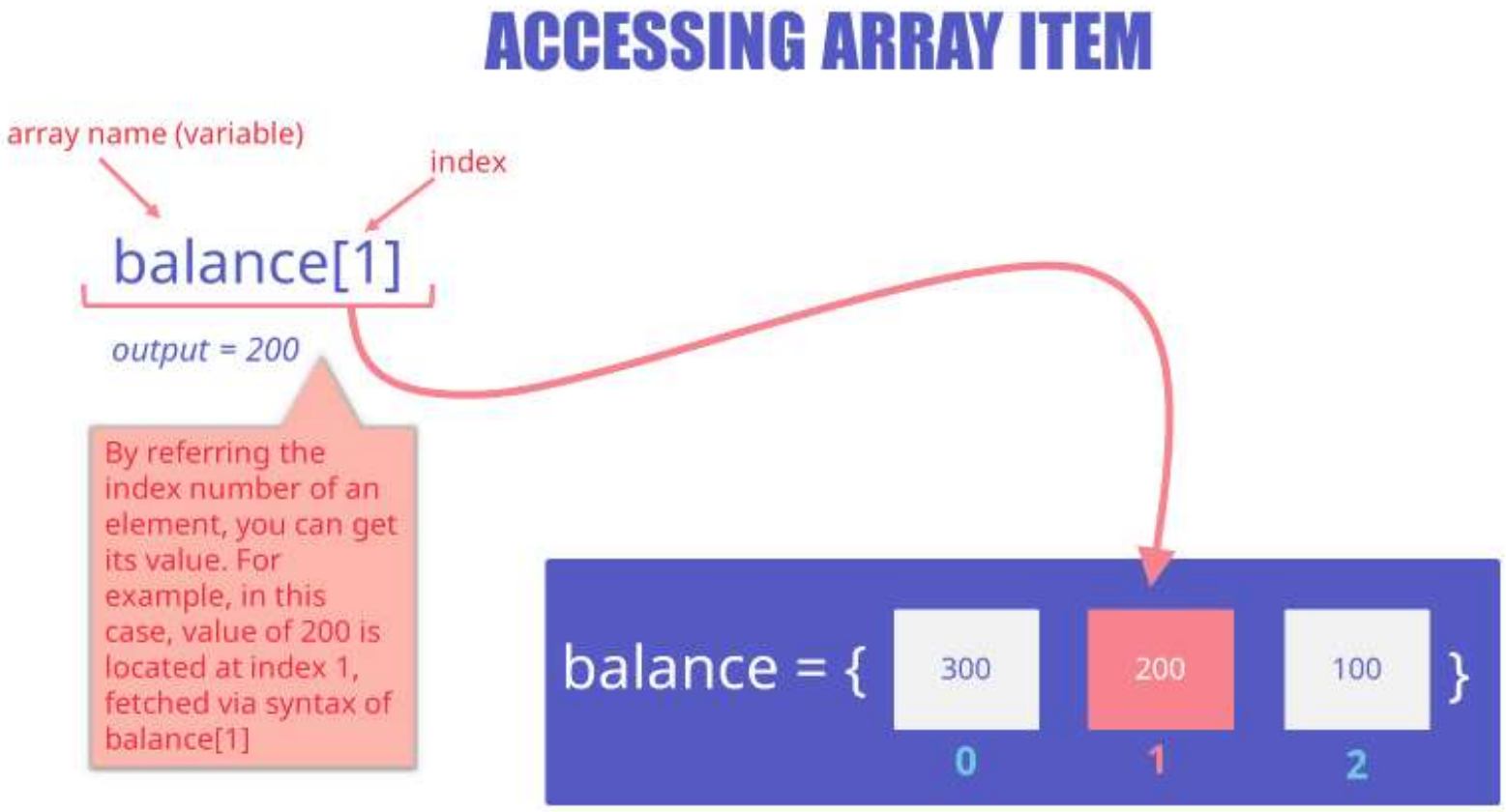
```
int balance[3] = { 300, 200, 100 }
```

data type — array name — elements



- **Array name:** necessary for easy reference to the collection of elements
- **Data Type:** necessary for type checking and data integrity
- **Elements:** these are the data values present in an array

- **Elements are stored at contiguous memory locations.**
- **An index is always less than the total number of array items.**
- **In terms of syntax, any variable that is declared as an array can store multiple values.**
- **Almost all languages have the same comprehension of arrays but have different ways of declaring and initializing them.**
- **However, three parts will always remain common in all the initializations, i.e., array name, elements, and the data type of elements.**



Here, we have accessed the second value of the array using its index, which is 1. The output of this will be 200, which is basically the second value of the balance array.

How to access a specific array value?

You can access any array item by using its index

Syntax

arrayName[indexNum]

Example

balance[1]

◦ Array Representation

◦ Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.

Thank You