



Marwadi
University
Marwadi Chandarana Group





01CE1301 – Data Structure

Unit - 1

Introduction to Data Structure

Prof. Megha Mudholkar

Department of Computer Engineering

- Data Management concepts
- Data types – primitive and non-primitive
- Types of Data Structure – linear and non-linear
- Abstract Data Types



Data Management Concepts

❑ What is Data?

- ❑ **Data** is the basic fact or entity that is utilized in calculation or manipulation.
- ❑ There are two different **types of data** - **Numeric** data and **Alphanumeric** data.
- ❑ When a programmer collects such type of data for **processing**, he would require **to store them in computer's main memory**.
- ❑ The process of storing data items in computer's main memory is called ***representation***.
- ❑ **Data** to be processed must be **organized in a particular fashion**, these organization leads to structuring of data, and hence the mission to study the Data Structures.





Data Types

❑ What is Data types?

- ❑ A **data type** is a classification of data, which can store a specific type of information.
- ❑ Data Type = Basic Data Type = Primitive Data Type

❑ Primitive Data Types

- ❑ A **primitive data** type is predefined by the language and is named by a reserved keyword.
- ❑ Primitive data types are predefined, supported by C language.
 - int, char, float, double.

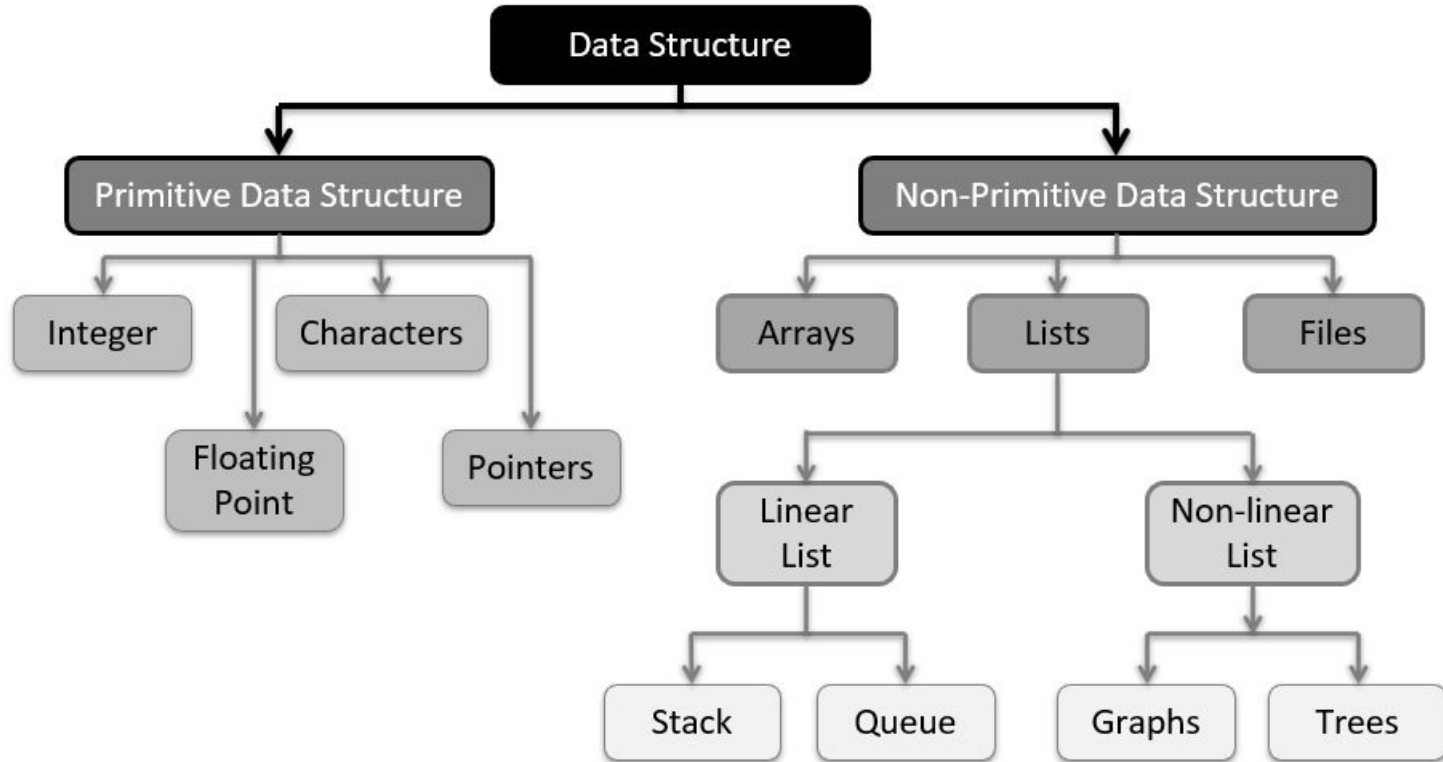
❑ Non-primitive Data Types

- ❑ **Non-Primitive** data types are not defined by C language, but are created by the programmer.
- ❑ They are created using the basic data types.
- ❑ Example:
 1. Linked List
 2. Stacks
 3. Queue
 4. Graph



Types of Data Structure

Types of Data Structure



□ Primitive data structures

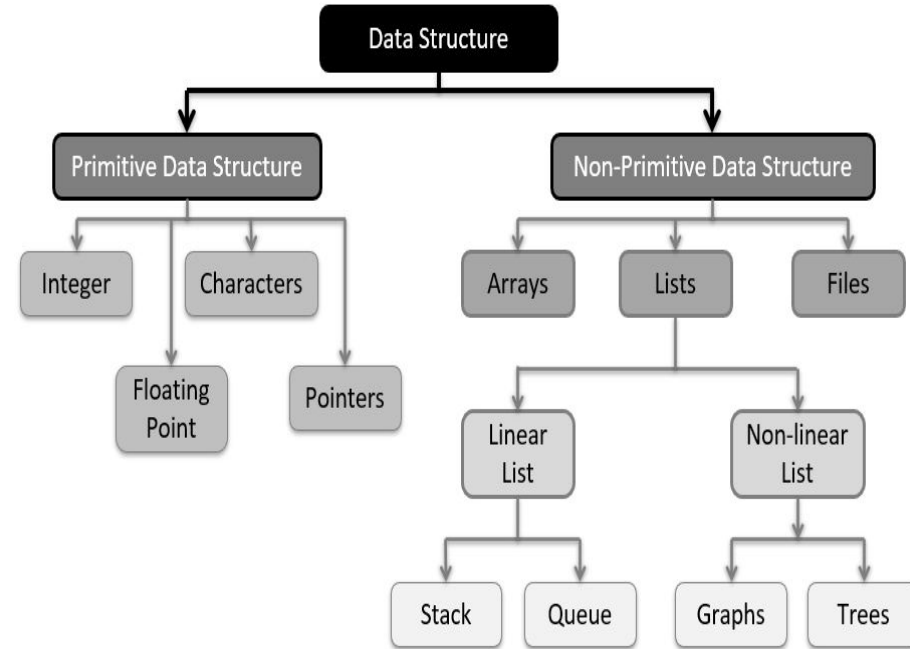
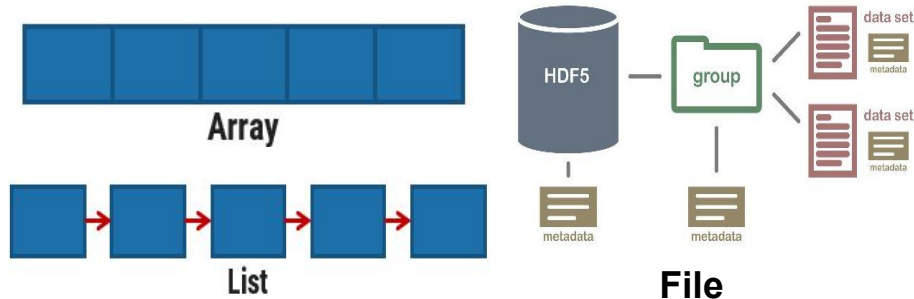
- Primitive data structures are basic structures and are directly operated upon by machine instructions.
- *Integers*, *floats*, *character* and *pointers* are examples of primitive data structures.

❑ Non primitive data structure

- ❑ These are derived from primitive data structures.
- ❑ The non-primitive data structures emphasize on structuring of a group of homogeneous or heterogeneous data items.
- ❑ Examples of Non-primitive data type are *Array*, *List*, and *File*.

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

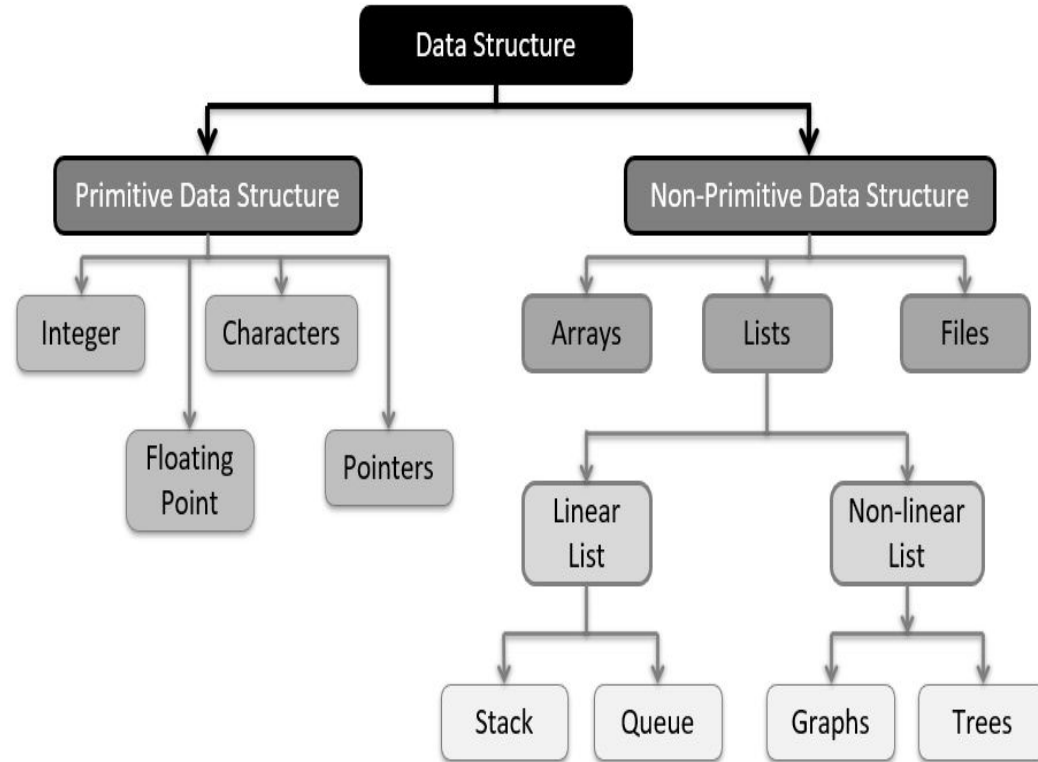
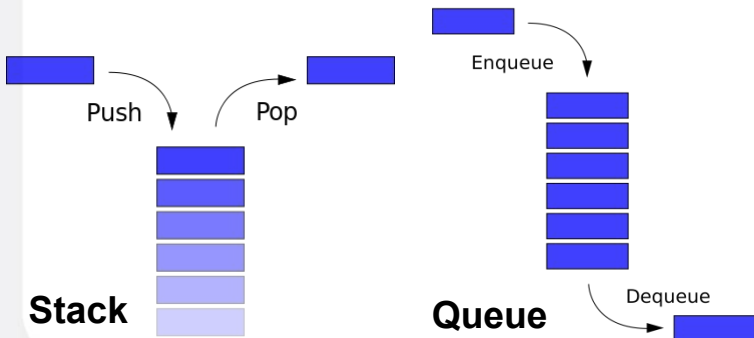


Linear Data Structure



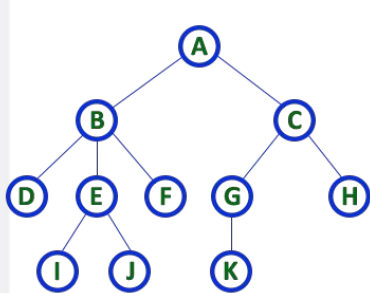
Linear data structures

- A data structure is said to be Linear, if its elements are connected in linear fashion by means of logically or in sequence memory locations.
- Examples of Linear Data Structure are **Stack** and **Queue**.

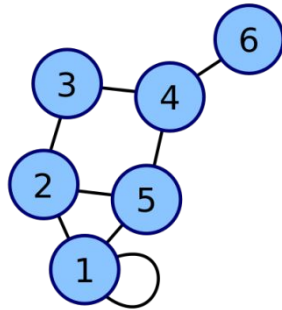


Nonlinear data structures

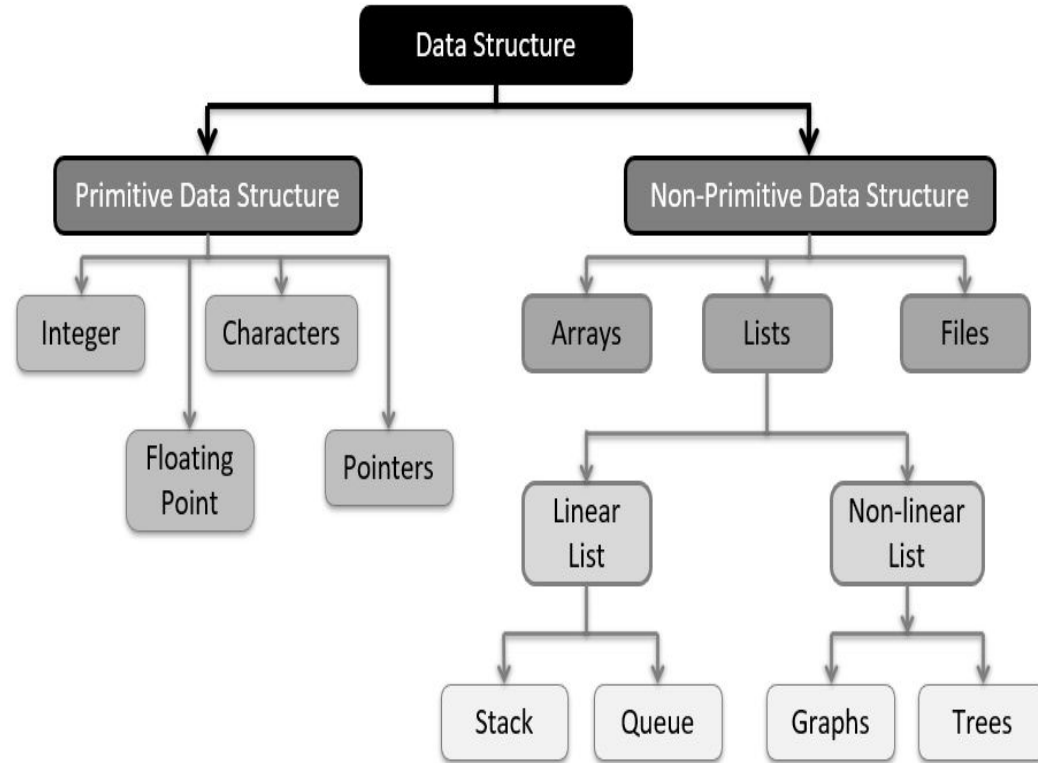
- Nonlinear data structures are those data structure in which data items are not arranged in a sequence.
- Examples of Non-linear Data Structure are **Tree** and **Graph**.



Tree



Graph





Abstract Data Types

- ***Abstract Data type (ADT)*** is a type (or class) for objects whose behavior is defined by a set of values and a set of operations.
- The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.
- It does not specify how data will be organized in memory and what algorithms will be used for implementing the operations. It is called “abstract” because it gives an implementation-independent view.

□ **Example:**

- Whenever end user uses Stack, he is concerned about only the type of data and operations that can be performed on it.
- Fundamentals of how the data is stored is invisible to users.
- They will have push() and pop() functions only.



Data Structure

❑ What is Data Structure?

- ❑ A data structure is a **way of organizing all data items** that **considers** not only the **elements stored** but also their **relationship to each other**.
- ❑ We can also define data structure as a **mathematical or logical model** of a particular **organization** of **data items**.

Operation of Data Structure



- ❑ **Create:** It results in reserving memory for program elements.
- ❑ **Destroy:** It destroys memory space allocated for specified data structure.
- ❑ **Selection:** It deals with accessing a particular data within a data structure.
- ❑ **Updation:** It updates or modifies the data in the data structure.
- ❑ **Searching:** It finds the presence of desired data item in the list of data items.
- ❑ **Sorting:** It is a process of arranging all data items in a data structure in a particular order.
- ❑ **Merging:** It is a process of combining the data items of two different sorted list into a single sorted list.
- ❑ **Splitting:** It is a process of partitioning single list to multiple list.
- ❑ **Traversal:** It is a process of visiting each and every node of a list in systematic manner.

THANK YOU

