UNIVERSITY OF DIYALA

Data Structure

Lecture 1:
An Introduction

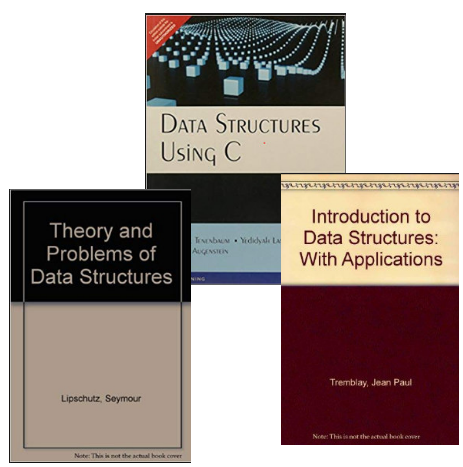
ASST. INSTRUCTOR
ALI A. AL-ANI

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

List Of Reference Material:

1. Tanenbaum Aaron M, Langsam Yedidyah, Augenstein J Moshe, **Data Structures using C**.
2. Tremblay J.P and Sorenson P.G, **An introduction to data structures with applications**, Tata McGraw Hill, 2nd Edition
3. Seymour Lipschutz, **Theory and Problems of data Structures**, Schaum's Series, Tata McGraw Hill, 2004.



Department of Computer Science
College of Science

Course Objectives

- ***Objectives:***

1. Encourages students to explore data structures by implementing them, a process through which students discover how data structures work and how they can be applied.
2. Students will be expected to write C++ language programs, ranging from very short programs to more elaborate systems.

- ***Pre-requisite:***

- A good knowledge of C++ language, use of Function and structures.

UNIVERSITY OF DIYALA

Introduction

- Representing information is fundamental to computer science. The primary purpose of most computer programs is not to perform calculations, but to store and retrieve information usually as fast as possible.
- For this reason, the study of data structures and the algorithms that manipulate them is at the heart of computer science. And that is what this subject is about helping us to understand how to structure information to support efficient processing.

Department of Computer Science
College of Science

Introduction

The main goals of studying the data structure concept are:

1. To present the commonly used data structures.

2. To introduce the idea of trade offs the concept that there are costs and benefits associated with every data structure. (space and time required for typical operations).

3. To learn how to measure the effectiveness of a data structure or algorithm. Only through such measurement can we determine which data structure in your toolkit is most appropriate for a new problem.

Data Structure Definition

- **In computer science, a data structure** is a particular way of storing and organizing data in a computer's memory (or sometimes on a disk) so that it can be used efficiently.
- ***Choice the format depends on two factor***

1.

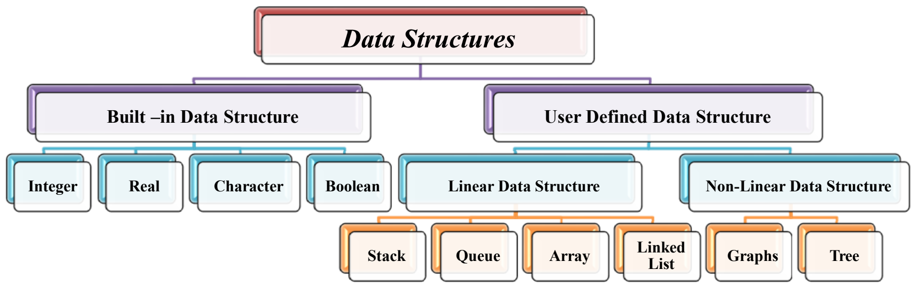
- ***It must be rich enough in structure to mirror the actual relationships of the data in the real world.***

2.

- ***The structure should be simple enough that one can effectively process the data when necessary.***

Classification of Data Structures

- Data Structure are generally classified into **Built-in** and **User defined** data structure. The **Built-in data type** consist of characters that cannot be divided such as (integer, real, character and Boolean). They are also called simple data type. The figure shows the classification of data structures.



How to choose the suitable data structure

1. ***Analyze your problem to determine the basic operations that must be supported. Examples of basic operations include inserting a data item into the data structure, deleting a data item from the data structure, and finding a specified data item.***
2. ***Data size and the required memory.***
3. ***The dynamic nature of the data.***
4. ***The required time to obtain any data element from the data structure.***
5. ***The programming approach and the algorithm that will be used to manipulate these data.***

UNIVERSITY OF DIYALA

User Defined Data Structure

1. Linear Data Structure.

- A data structure is said to be linear if its elements form a sequence, or, in other words, a linear list.
- ***There are two basic ways of representing such linear structures in memory:***

1.

- *One way is to have the linear relationship between the elements represented by means of sequential memory locations. these linear structures are called **Arrays**.*

2.

- *The other way is to have linear relationship between the elements represented by means of **pointers** or **links**. These linear structures are called **Linked List**.*

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

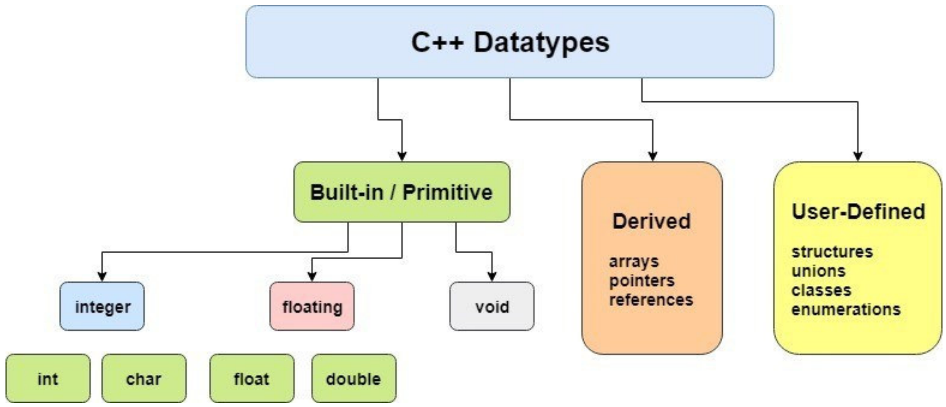
Linear Data Structure

*The operations that normally performs on any linear structure, whether it be an **array** or a **linked list**, include the following:*

- ***Traversing:** means to visit all the elements of the array in an operation is called traversing.*
- ***Sorting:** is the Re-arrangement of values in a list in a specific order (Ascending / Descending).*
- ***Searching:** The process of finding the location of a particular element in a list is called searching. There are two popular searching techniques/mechanisms : Linear search and binary search*
- ***Insertion:** Adding a new element to the list.*
- ***Deletion:** Removing an element from the list.*
- ***Merging:** Combining two lists into a single list.*

Department of Computer Science
College of Science

Data Types in C++



UNIVERSITY OF DIYALA

Linear Data Structure: Array

- The array is probably the most widely used data structure; in some languages it is even the only one available.
- **An array** is a collection of elements of the same type, called its base type; it is therefore called a **homogeneous structure**.
- The array is a random-access structure, because all components can be selected at random and are equally quickly accessible. Array is very useful data structure provided in programming language.

However, it has at least two limitations:

1. *Its size has to be known at compilation time.*
2. *The data in the array are separated in computer memory by the same distance, which means that inserting an item inside the array requires shifting other data in this array.*

Department of Computer Science
College of Science

Linear Data Structure: Array

Array: Classification

- ***One – Dimensional Arrays***
- ***Two– Dimensional Arrays***
- ***Multidimensional Arrays***

UNIVERSITY OF DIYALA

One – Dimensional Arrays

- A one dimensional array is used when it is necessary to Keep a large number of items in memory and reference all the items in uniform manner.
- **Int N[100];**
- That is mean we reserves **100** successive memory locations and each location is large enough to conation single integer. In C++, the array element indices are 0-99.

N[0]	N[1]	...	...	...	...	...	...	...	N[99]

Department of Computer Science
College of Science

One – Dimensional Arrays

- The address of the first location is called the **base address** of the array and is denoted by base (**BA**) and the rest of the array elements come after this address.
- Computer does not need to keep track of the address of every array element, but need to track only the address of the first element of the array **Base Address (BA)** and to reach to any array element and the compiler use the following formula to do so.

$$Loc\ (\ N\ [I]\ ) = BA + (\ I\ ) \times Size$$

- **Loc N[I] :** The location of the element **I**, **BA**: Fixed base address, **Size:** A fixed constant, is also known as size of the data type.

UNIVERSITY OF DIYALA

One – Dimensional Arrays: Example

- **Example :** Consider an one dimension array (N) with size 10 and the base address equal (3002) and each element of the array occupy 1 byte. find the address the element number six.
- ***Loc (N [I]) = BA + (I) × Size***
- ***then, Loc (N [5]) = 3002 + (5) × 1***
- ***Loc (N [5]) =3007.***

The Physical Representation Of Array In Memory

Logical address	Physical address	Memory
N[0]	3002	
N[1]	3003	
N[2]	3004	
N[3]	3005	
N[4]	3006	
N[5]	**3007**	
:	:	
N[9]	3011	
Programmer view	Compiler view	

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

One – Dimensional Arrays: Example

- **Example 2:** Consider an one dimension array (N) with size 30 and the base address equal (200) and each element of the array occupy 2 byte. find the address the element number 16.
- ***Loc (N [I]) = BA + (I) × Size***
- *then,* ***Loc (N [15]) = 200 + (15) × 2***
- ***Loc (N [15]) =230.***

The Physical Representation Of Array In Memory

Logical address	Physical address	Memory
N[0]	200	
	201	
:	:	
N[15]	**230**	
	231	
:	:	
N[30]	260	
	261	

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Two – Dimensional Arrays

- **2D array** is a data structure type that consists of a set of elements that are all of the same type, and all elements are distributed on a set of rows and columns that represent the size of the array.
- **Int N[3][5];** That is mean we reserves (3*5=15) successive memory locations and each location is large enough to conation single integer. The number of Rows or Columns is called the range of the dimension.
- The array N will be represented in the memory by block of (3 × 5) sequential memory location. Programming language will store array N either :

  1. ***Column by Column****: called (Column-Major Order) Ex: Fortran, Matlab.*
  2. ***Row by Row****: called (Row-Major Order) Ex: C, C++, Java.*

Department of Computer Science
College of Science

Two – Dimensional Arrays

- ***By Column N[2][3].*** *{(1,5,3) , (4,8,6)}*

	Column 0		Column 1		Column 2	
value	1	4	5	8	3	6
address	100	101	102	103	104	105

- ***By Column N[2][3].*** *{(1,5,3) , (4,8,6)}*

	Row 0			Row 1		
value	1	5	3	4	8	6
address	100	101	102	103	104	105

UNIVERSITY OF DIYALA

Two – Dimensional Arrays

1. ***Column-Major Order :***

- ***Loc (N [I][J]) = BA + (m × J + I) × Size***
- ***Where:***
  - ***Loc N[I][J] :*** *the location of the element* ***I,J****.*
  - ***BA:*** *fixed base address.*
  - ***m:*** *number of row.*
  - ***Size:*** *a fixed constant, is also known as size of the data type.*

UNIVERSITY OF DIYALA

Column-Major Order : Example

- **Example:** Consider a two dimension array (N) with size (m=3 × n= 5) and the base address equal (300) and each element of the array occupy 1 byte. find the address the element N[1][2].Suppose the programming store 2D using Column-Major.
  - ***Loc (N [I][J]) = BA + (m × J + I) × Size***
  - ***Loc (N [1][2]) = 300 + (3 × 2 + 1) × 1***
  - ***Loc (N [I][J]) = 307***

	L.A.	P.A.	Memory
Col 0	N[0][0]	300	
	:	:	
Col 1	N[0][1]	303	
	:	:	
Col 2	N[0][2]	306	
	N[1][2]	**307**	
	N[2][2]	308	
Col 3	N[0][3]		
	:	:	
Col 4	N[0][4]		
	:	:	

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Two – Dimensional Arrays

2. ***Row-Major Order :***

- ***Loc (N [I][J]) = BA + (n × I + J) × Size***
- ***Where:***
  - ***Loc N[I][J] :*** *the location of the element* ***I,J****.*
  - ***BA:*** *fixed base address.*
  - ***n:*** *number of column.*
  - ***Size:*** *a fixed constant, is also known as size of the data type.*

Department of Computer Science
College of Science

Row-Major Order :Example

- **Example :** Consider a two dimension array (N) with size (m=3 × n= 5) and the base address equal (600) and each element of the array occupy 1 byte. find the address the element N[2][3].Suppose the programming store 2D using Row-Major.
- ***Loc (N [I][J]) = BA + (n × I + J) × Size***
- ***Loc (N [2][3]) = 600 + (5 × 2 + 3) × 1***
- ***Loc (N [I][J]) = 613***

	L.A.	P.A	Memory
Row 0	N[0][0]	600	
	:	:	
Row 1	N[1][0]	605	
	:	:	
Row 2	N[2][0]	610	
	N[2][1]	611	
	N[2][2]	612	
	N[2][3]	**613**	
	N[2][4]	614	

UNIVERSITY OF DIYALA

Linear Data Structure: Array

- ***H.W.1:*** *Consider Int X[3][4] ,what is the address of the X[2][2] if the base address equal (300) and each element of the array occupy 1 byte. Suppose the programming store 2D using Column-Major.*
- ***H.W.2:*** *Consider Int X[5][3] ,what is the address of the X[3][2] if the base address equal (100) and each element of the array occupy 2 byte. Suppose the programming store 2D using Row-Major.*

The End