CS101
Data Structures and Algorithms

Lecture 02

Data Types

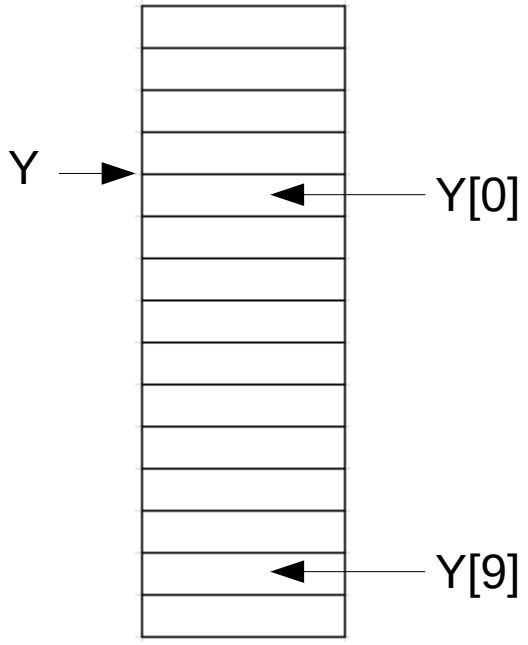
Generic Data Types -1

- Atomic / Basic Data types
  - Char / Byte – 1 Byte
  - Integer / Short – 2 bytes / Long – 4 bytes / long long – 8 bytes
  - Float – 4 bytes / Double – 8 bytes (IEEE 754 representation)
  - Enum – similar to ordinal type : No direct h/w representation
- Byte / Integer / Float / Double have direct machine representation
- Enum is programming language abstraction
  - Can mean a byte / short / long

Data Type Constructor

- Homogeneous
  - Array : Eg. Array of integers, String – Array of chars (ASCII) – (UNICODE is multibyte character encoding)
  - List : Eg., list of adjacencies of a node in a graph
- Heterogeneous – Record : Eg., Library Book details, Student details
- Note: Memory is Linear -> One dimensional.
  - Successive memory locations have consecutive addresses:

Array

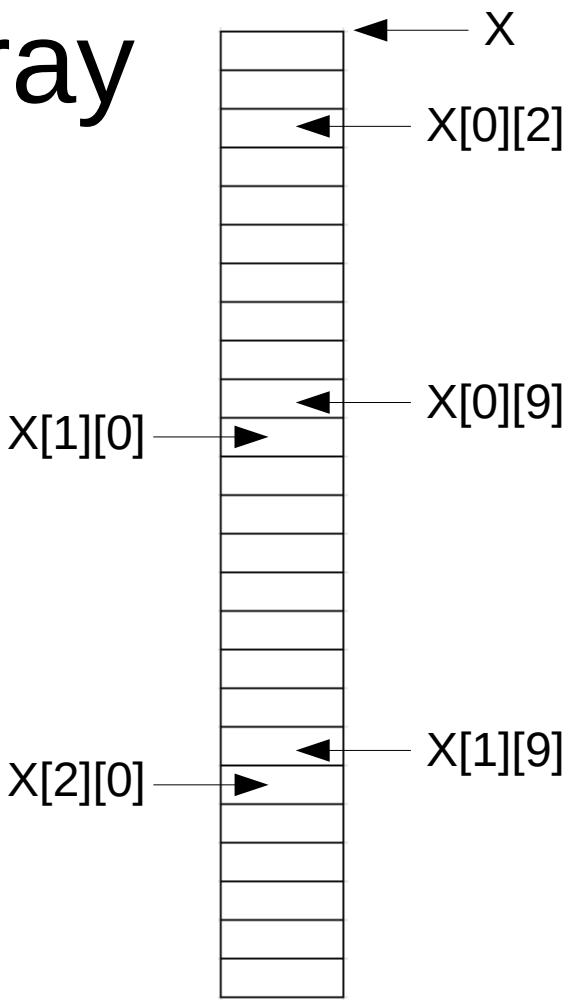


- Single Dimension Array
- Eg., Y[100] – An array Y of 100 elements
- Arrays permit indexed access.
  - Y is the starting address of the array
  - Y[0] is the first element, Y[9] is the 10nth element and so on.
- Once allocated, the array extent (size and location) is fixed

Caveat: In C/C++ the array extents can be modified because of dynamic allocation

Multidimensional Array

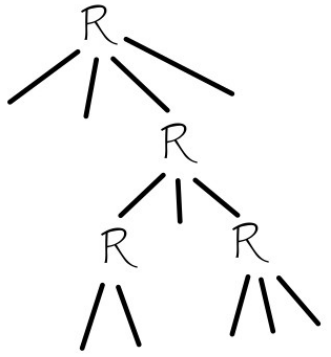
- X[5][10] has 5 rows and 10 columns :
- X is the starting address of the 2D-array
  - Row Major layout
    - Row elements occupy consecutive addresses in memory
    - X[2][3] element is (X+2*10+3)
  - Column Major layout
    - Column elements occupy consecutive addresses in memory
    - X[2][3] element is (X+2*5+3)



Row major layout

Record

- Composite Data Structure: Record is a data type constructed from other data
- Eg., Student Record contains following data (also called Fields)
  - Name of the Student : String type
  - Registration No: String type
  - Date of Birth : Date type (this is again another Record type)
  - Gender : Character type
  - Degree program : Enum
- Design the record structure for a Library Catalogue



Records are
Hierarchical data structures

Python Data Structures

- **Strings** – `"hello world\\n" – r"Hello world\n"`
- List – `[0,3,[4,5],3,"Hello",33,40.4]`
- **Tuples** – `(0,3,[4,5],3,"Hello",2,40.4)`
- **Set** – `{0,3,(4,5),3,"Hello",33,40.4}`
- Dict – `{'a':'Some value', 'b':[1,2,3,4]}`
- **Boolean** – `True, False`

Pages 7-10 in The Book

Python Data Structures

- Int – float – str – tuples – bool – are immutable
  - For x,y,z integers, what happens in $x = x + y + z$?
  - The values of x, y, z are added and the result is renamed as x (not assigned to x)
- List – Dict – Set – are Mutable

Pages 7-10 in The Book

A note on Class vs Object

- Class model Data and Operation (method / Interfaces)
- Objects are runtime instances of Class
- List in python is a class
- `X = [0,3,[4,5],3,"Hello",33,40.4]` is a list object (or an instance of list)
- 0, 3, "Hello" etc. constitutes data in the list object
- What are the list methods?

Methods in list class -1

- append() Adds an element at the end of the list
- clear() Removes all the elements from the list
- copy() Returns a copy of the list
- count() Returns the number of elements with the specified value
- extend() Add the elements of a list (or any iterable), to the end of the current list

Methods in list class -2

- index() Returns the index of the first element with the specified value
- insert() Adds an element at the specified position
- pop() Removes the element at the specified position
- remove() Removes the first item with the specified value
- reverse() Reverses the order of the list
- sort() Sorts the list

Methods in String class ??