Course code : **SWE2001**
Course title : **Data Structures and its Application**

Introduction to Data Structures

Introduction

Data

- A collection of facts, concepts, figures, observation, occurrences or instructions in a formalized manner.

Information

- The meaning that is currently assigned to data by means of the conventions applied to those data (i.e processed data).

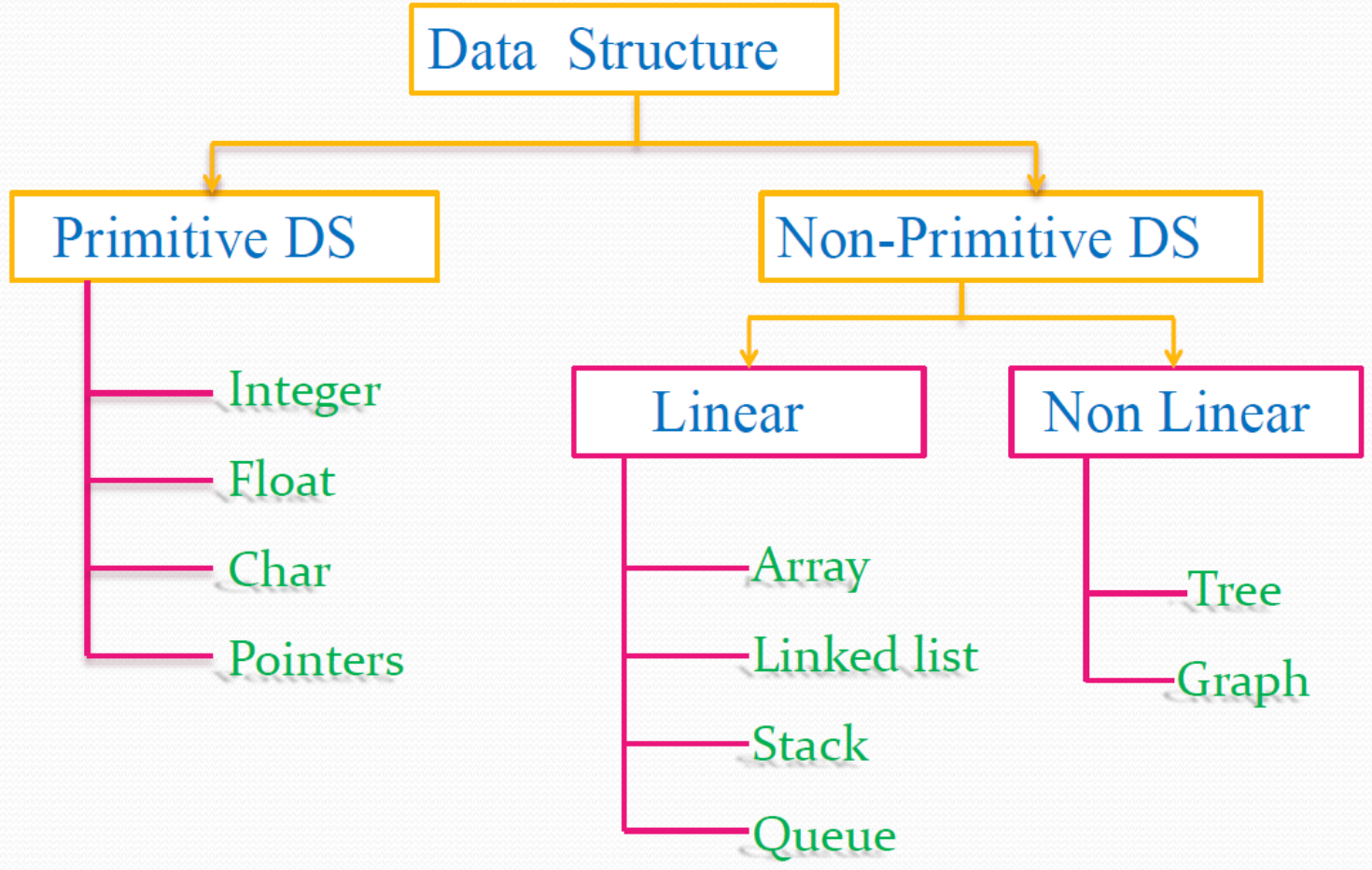
Data Type

- Set of elements that share common set of properties used to solve a program.

Data Structure

- Data structure describes *the way of storing and organizing the data*.

Classification



Abstract Data Types (Non Primitive)

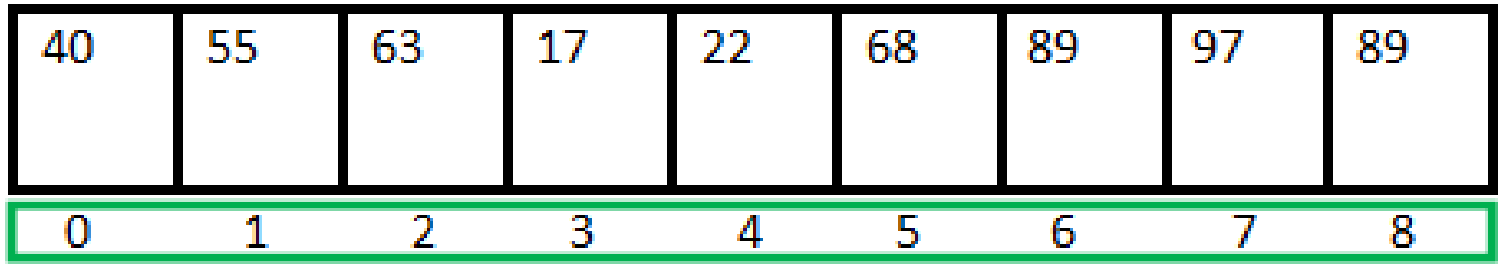
Abstract Data type (ADT) is a type (or class) for objects whose *behavior is defined by a set of value and a set of operations*.

Types of operations in ADT

- **Traversing** - Accessing and processing the records at once.
- **Searching** - Finding the location of a data.
- **Inserting** - Inserting a new data to data structure.
- **Deleting** - deleting a new data to data structure.
- **Sorting** - Arranging the given elements in some order.
- **Merging** - Combining the values of two data structure into one data structure.

Arrays

An array is a group of similar-type variables that are referred to by a common name.



Array Length = 9
First Index = 0
Last Index = 8

Creating Arrays

- An array is a container object that *holds a fixed number of values of a single type*
- When an array is created, *the length of an array is fixed*
- *Array elements are automatically initialized* with the default value of their type, When an array is created
- Array can be created using the new keyword
- **Ex:**
    - int[] x = new int[5]; // defining an integer array for 5 blocks

One Dimensional Arrays

- Alternatively, we can create and initialize array as below format

```
int[] x = {10, 20, 30};
int[] x = new int[]{10, 20, 30};
```

- Here the length of an array is determined by the number of values provided between { and }
- The built-in length property determines the size of any array

Ex:

- int[] x = new int[10]; int x_len = x.length;

Array - Example

```
public class ArrayDemo {
	public static void main(String[] args)
	{
		int[] x; // declares an array of integers
		x = new int[5]; // allocates memory for 5integers
		x[0] = 11;
		x[4] = 22;
		System.out.println("Element at index 0: " + x[0]);
		System.out.println("Element at index 1: " + x[1]);
		System.out.println("Element at index 4: " + x[4]);
	}
}
```

Two / Multi dimensional Array

- Two-dimensional arrays are arrays of arrays
- Initializing two-dimensional arrays:

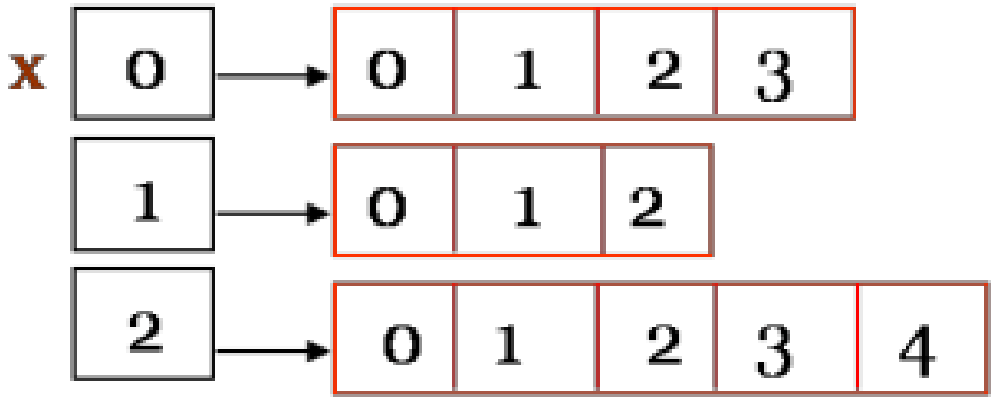
  int[][] y = new int[3][3];

  The 1st dimension represent rows or number of one dimension, the 2nd dimension represent columns or number of elements in the each one dimensions

- The curly braces { } may also be used to initialize two dimensional arrays
- Ex:
  - int[][] y = { {1,2,3}, {4,5,6}, {7,8,9} };
  - int[][] y = new int[][] { {1,2,3}, {4,5,6}, {7,8,9} };

2D Array example

- **Ex2:**
  - int [][]x = new int[3][];
  - x[0] = new int[]{0,1,2,3};
  - x[1] = new int []{0,1,2}; x[2] = new int[]{0,1,2,3,4};



Array Operations

- Insert
- Delete
- Search
- Sort
- Display

Insert

```
public static void insert(int pos,int ele)
{
        for (int i = size - 1; i > pos; i--) {
                ary[i] = ary[i - 1];
        }
        ary[pos] = ele;
}
```

Delete

```
public static void aryDelete(int pos) {
int temp[]=new int[size-1];
for (int i = 0, k = 0; i < ary.length; i++) {
            if (i == pos) {
                continue;
            }
            temp[k++] = ary[i];
        }
ary=temp;
}
```

Sorting

```
public static void sort() {
        int pass=0;
        for(int i=0;i<ary.length-1;i++) {
                    if(ary[i]>ary[i+1]){
                        int temp=ary[i];
                        ary[i]=ary[i+1];
                        ary[i+1]=temp;
                        pass=1;
                    }
        }
        if(pass==1)
        sort();
}
```

Display

```
public static void display() {
    for (int i : ary)
        System.out.println(i);
}
```