Data Structure Basics

&

Concepts of

Abstract Data Types

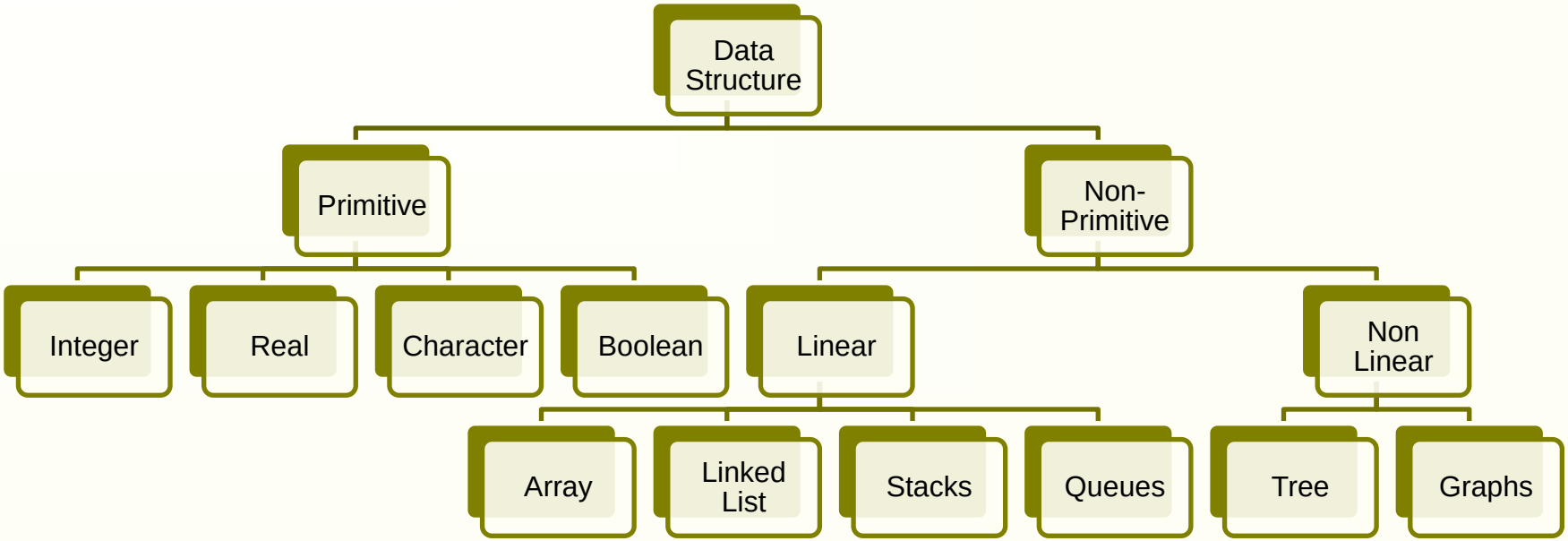
Data Structure

The logical or mathematical model of a particular organization of data structure

a) Data Structure should be **RICH ENOUGH** in structure to mirror the actual relationships of the data in the real world

b) Data structure should be **SIMPLE ENOUGH** that one can efficiently process the data

Classification of Data Structure



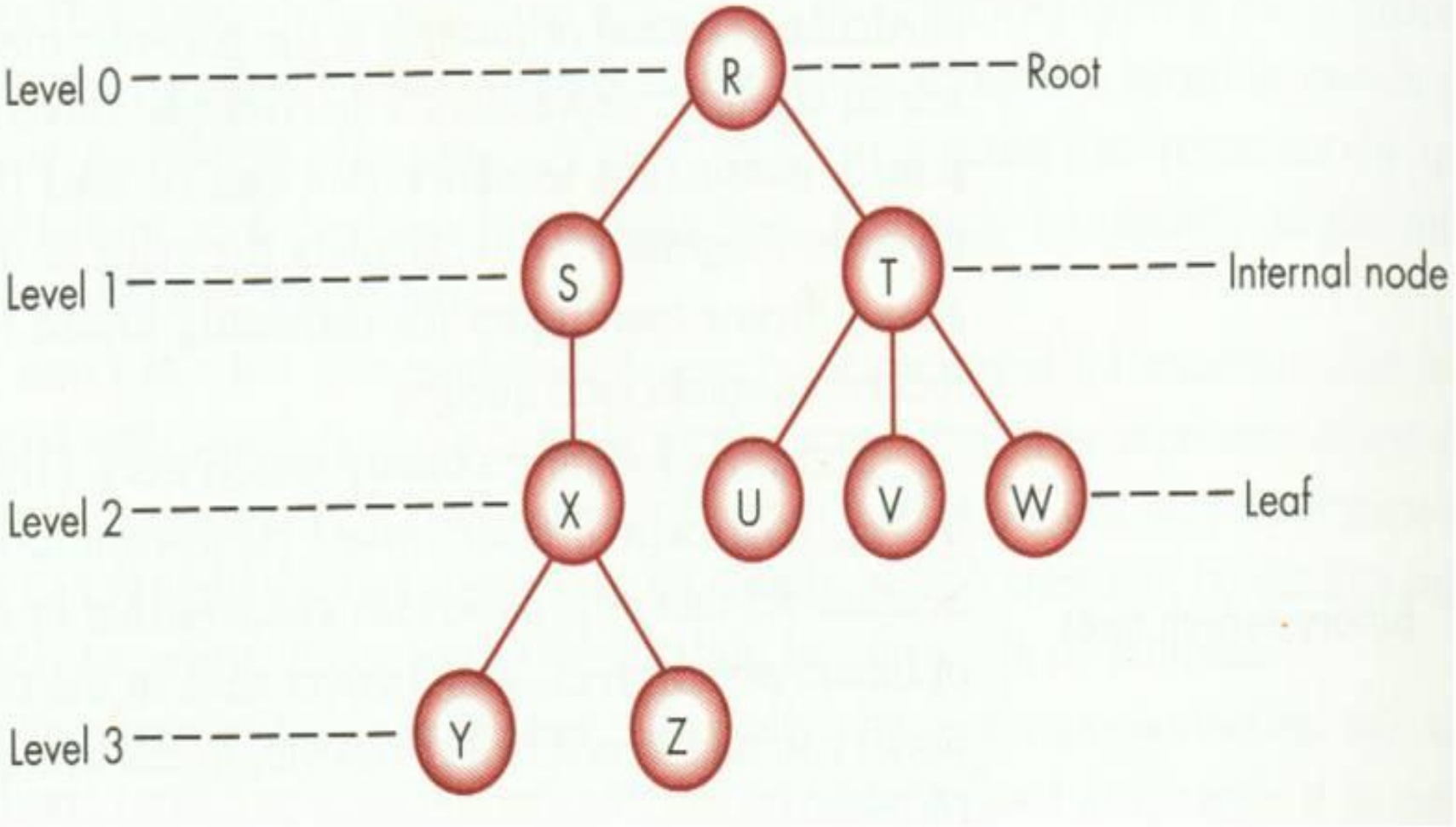
Data Structure: Example

- Array
- Linked List
- Trees
- Graphs

Data Structure: LINKED LIST

Sl.No.	SUBJECT NAME	FACULTY NAME
1	Data Structure	Rajeev
2	Computer Network	Arnab
3	DBMS	Rajesh
4	CPP	Rajeev
5	Oracle	Ankur
6	Architecture	Ramesh
7	Soft Computing	Arnab
8	Economics	Ankur
9	Basic Electronics	Ram
10	Basic Electrical	Ram

Data Structure: LINKED LIST



Abstract Data Type

- ADT is a tools for specifying the logical properties of a data type
- Definition of a data type WITHOUT any connection to the actual implementation in a programming language.
- A mathematical abstraction of a data structure.

The actual IMPLEMENTATION can be done by the programmer.

Two Parts:

What does it look like? Structural Definition

What does it do? Functional Definition

ADT: List Representation

- Linear List
- Matrix
- Tree
- Graph

An Example ADT - Employee

- **ADT Employee**
  - **Employee Information** - Name, Employee Number, Gender, DOB, DOJ
  - **Functions** - join, work, travel, resign, retire

An Example ADT - Fraction

- **ADT Fraction**
  - **Data** - Numerator, Denominator
  - **Functions** - add, subtract, divide, multiply, reciprocal, etc.

An Example ADT - Complex Numbers

- **ADT Complex**
  - **Data** - Real Part, Imaginary Part
  - **Functions** - add, subtract, divide, multiply, etc.

ADT - Properties

- During defining ADT as mathematical concept, we are not concerned about
  - SPACE
  - TIME EFFICIENCY

ADT of RATIONAL NUMBER

An Example ADT - One Dimensional Array

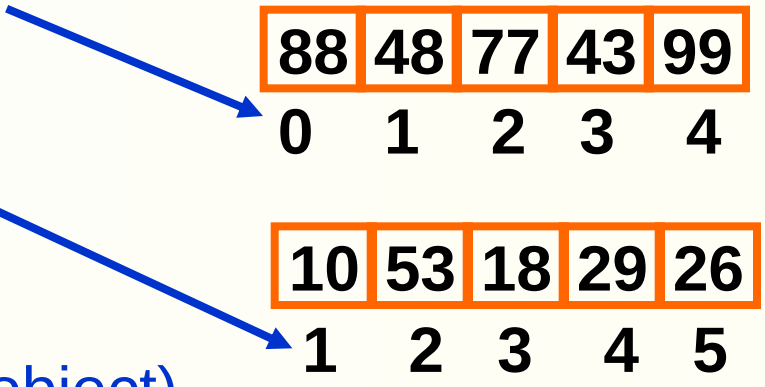
Array

- An ordered set of similar objects (same type).
- Fixed number of objects.
- Each object is assigned an index number.

0-based (0,1,2,3,…,N-1)

OR

1-based (1,2,3,…,N)



Two operations are defined:

store (arrayname,index, object)

object retrieve (arrayname,index)

Array

A more complete set of operations could be:

store (arrayname, index, object)

object retrieve (arrayname,index)

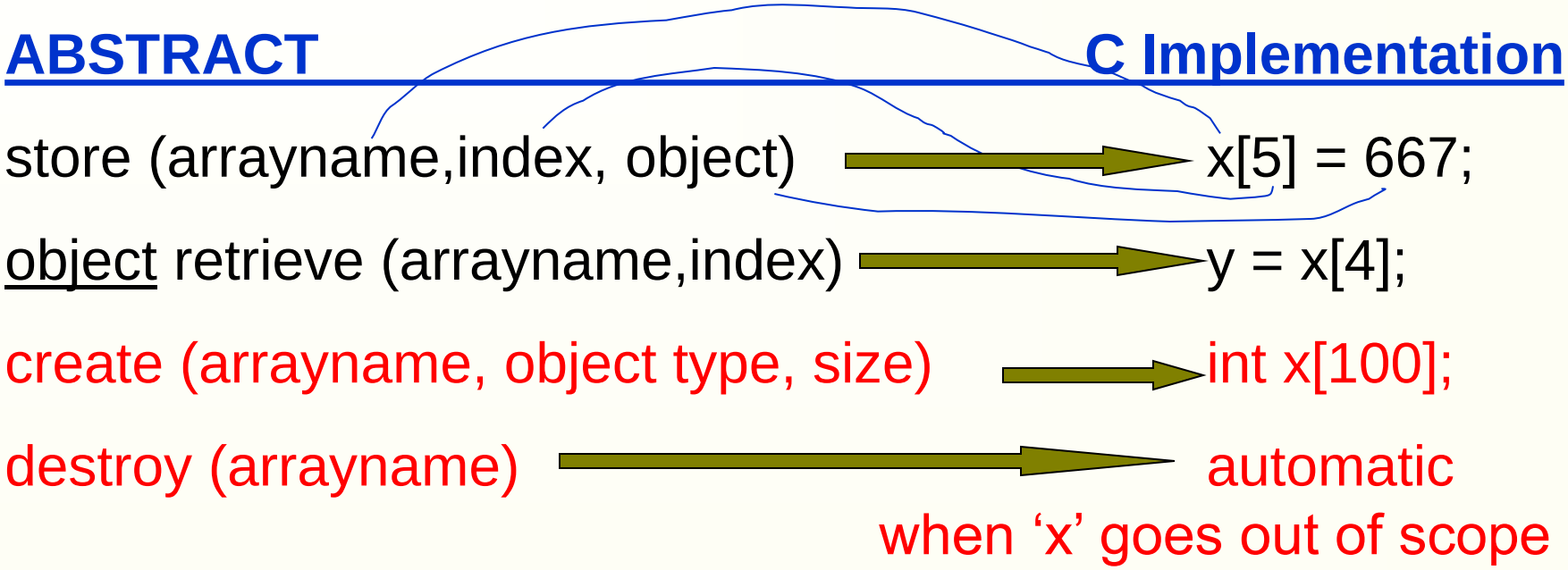
create (arrayname, object type, size)

destroy (arrayname)

Array - Implementation in C

In built data structure in C.

User defined implementation is not required.



Array - Implementation in C

```
typedef double ITEMTYPE; //array of fixed type
typedef struct
    {
     ITEMTYPE * arrptr;
    }ARRAY;
```

Array - Implementation in C

```
void create (ARRAY * x, int size);

void store (ARRAY * x, int index, ITEMTYPE obj);

ITEMTYPE retrieve (ARRAY * x, int index);

void destroy (ARRAY * x);
```

Array - Implementation in C

```
void create (ARRAY * x, int size)
{
x->arrptr = (ITEMTYPE *)malloc(size * sizeof(ITEMTYPE));
}
void store (ARRAY * x, int index, ITEMTYPE obj)
{
*(x->arrptr+index) = obj;
}
```

Array - Implementation in C

```
ITEMTYPE retrieve (ARRAY * x, int index)
        {
                return *(x->arrptr+index);
        }
void destroy (ARRAY * x)
        {
                free((void *)x->arrptr);
        }
```

Array - Implementation in C

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
        ARRAY  x;
        double pp=55.6;
        create (&x,100);
        store(&x,2,4.5);
        store(&x,0,pp);
        printf("%lg\n",retrieve(&x,220));
        printf("%lg\n",retrieve(&x,2));
        destroy(&x);
         return 0;
}
```

Abstract Data Types

LINEAR STRUCTURES

Linked Lists (4 types)

Linear and Circular

Singly linked and Doubly linked

2 implementations - DMA and array based

Abstract Data Types

LINEAR DATA STRUCTURES

Array

- One Dimensional Array
- Matrices – Two Dimensional Implementation
- Sparse Matrices

Abstract Data Types

LINEAR DATA STRUCTURES

Stacks

- Array Based Implementation
- Linked List Based Implementation

Queues

- Array Based Implementation
- Linked List Based Implementation

Abstract Data Types

NONLINEAR DATA STRUCTURES

Trees

- Binary Trees
- AVL Trees
- B-Trees
- General Trees and Forests

Abstract Data Types

NONLINEAR STRUCTURES

Graphs

Directed and Undirected

Weighted and Unweighted

THANK YOU!!