

2D Arrays

Monday, 20 March 2023 3:18 PM

Variables

ID: []
Name: []
Class: []
Sub: []
Fee: []

Since the array may have only one data type defined for all elements; we have to choose String that fits in for all other data types.

Column index	0	1	2	3	4
0	1	ZAFAR	AIS2	CMP101, CHEM	15000
1					
2					
3					
4					
5					
6					

Rows index

Array Operations:

1. Declaration : Asking computer to create space for the array in memory.
2. Initialisation : To clean garbage all elements of array are set to a value.
3. Population : Taking input from user & storing it in subsequent array elements.
4. Output : A method/Procedure to output all array elements.
5. Linear Search : A particular method; that searches line by line in array.
6. Bubble Sort : A particular algorithm to sort all array items.

DECLARATION:

Syntax: DECLARE <arrayName> : ARRAY [LB:UB, LB:UB] OF <datatype>

Statement: DECLARE StuRec : ARRAY[0:6, 0:4] OF STRING

INITIALISATION:

```
FOR Row ← 0 TO 6  
    FOR COL ← 0 TO 4  
        myArr[Row, COL] ← ""  
    NEXT COL
```

NEXT Row

Initialisation default values:

Integer/Real/Currency ← 0
STRING ← "" → Null
CHAR ← '' → Null
BOOLEAN FALSE

POPULATION: Array population is a process of filling an array with data values. It involves initialising the array with its initial values to the array elements. Population can be done in various ways, such as initialising the array with predefined values, assigning values within a loop or series of statements and reading values from a file or user input.

// input data from keyboard and store in subsequent array 11 elements

```
FOR Row ← 0 TO 6  
    INPUT "ID:", ID  
    INPUT "Student Name:", sName  
    INPUT "Student class:", sClass  
    INPUT "Student Subjects:", sSub  
    INPUT "Student Fee:", sFee  
    myArr[Row, 0] ← ID  
    myArr[Row, 1] ← sName  
    myArr[Row, 2] ← sClass  
    myArr[Row, 3] ← sSub  
    myArr[Row, 4] ← sFee
```

NEXT.

OUTPUT: An algorithm that reads (by traversing in order) all array elements and output them to the screen or the printer.

```
FOR R ← 0 TO 6  
    FOR C ← 0 TO 4  
        OUTPUT myArr[R, C]
```

NEXT C

NEXT R

You can take input directly in required array element.