Assignment -7

31-08-2023

Q1] What is meant by an array? Explain the use of array with escample.

Ans :) Array is an linear data structure
:) Array is considered as derived data type in C, C++, Java
:) Array is a derived data type which holds multiple homogeouns elements in indexed format.

:] When we store elements in primitive data type, each variable must have a seperate name & seperate memory. this is a long task to store many variables.

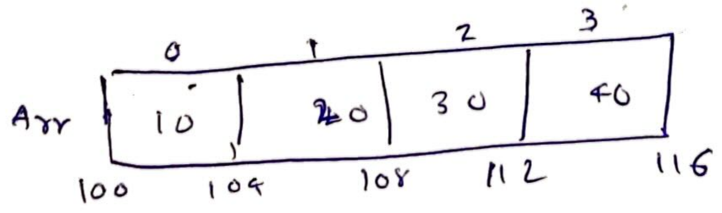
```
main {
      int no1=10;
      int no2= 20;
      int no3 =30;
      int no4=40;
     }
```

:] But in case of array a sequential memory is allocated for array.
:] In array there is no need to remember seperate names as you can access the array using a single name.
:] Each element of array has a unique index number.
:] In C, C++ &Java indexing of array starts from '0'

Example :

int Arr [4] = {10, 20, 30, 40};

This sentence is read as;
Arr is a dimensional array, consisting of 4 elemente, each element of type integer.

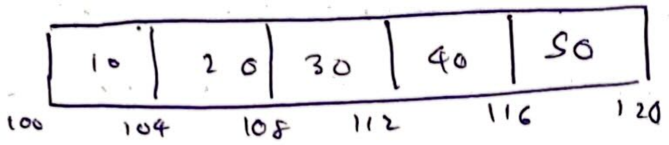


Q2] ~~Then~~ Whate are the types of array?

Anes There are two types of arrays based on dimensions:

1] <u>One - dimensional array</u> :

int arr[5] = { 10, 20, 30, 40, 50 };



2] <u>Multi - dimensional Array</u> : They are arrays of arrays

Two - dimensional Array can be created as:

int arr[3][2] where 3 is the number of rows
& 2 is the number of columns

<u>It is read as:</u>

arr is a two-dimensional array, which contains three one dimensional arrays, each one dimensional array contains 2 elements, each of type integer.

Q3] What are the ways in which we can initialise the elements of array?

Ans There are 3 ways :

1] ~~int~~ char arr[] = {'A', 'B', 'C', 'D'}
2] char arr[4] = {'A', 'X', 'C', 'Y'}

} Member initialisation List

3] char arr[4];

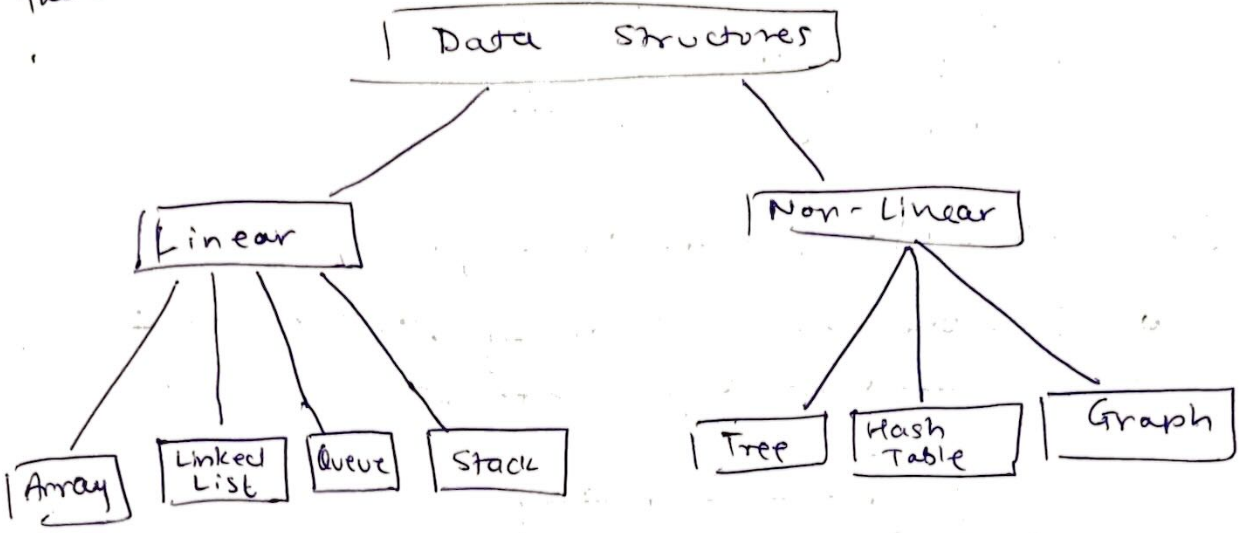
char arr[0] = 'A';
char arr[1] = 'B';
char arr[2] = 'C';
char arr[3] = 'D';

} Member by Member Initialisation List

Q4] What is meant by data structures? And what are types of data structures?

Ans Data structures is a way of storing & representing data in a particular format.

There are two types of data structures



Q5] What are the ways in which we can allocate memory for an array?

Ans We can allocate memory for array in 2 ways

1] Member initialisation

2] Member by member initialisation

1] Member initialisation by specifying the size of array

```
int Arr[3] = {10,20,30};
```

2] Member initialisation by not specifying size of array

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

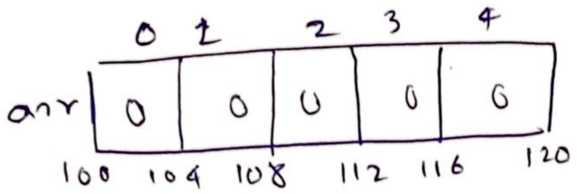
3] Member by member initialisation

```
int Arr[3];
~~A~~rr[0] = 10;
Arr[1] = 20;
Arr[2] = 30;
```

Q6] Read the below statements & draw its diagrammatic representation.

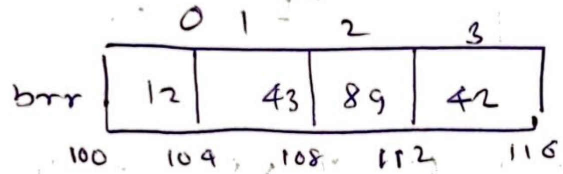
1] int arr[5];

arr is a one dimensional array, which has 5 elements, each element of type integer.



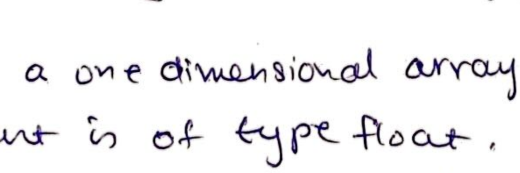
2] int brr[4] = {12, 43, 89, 42};

brr is a one dimensional array, which has 4 elements, each element of type integer.



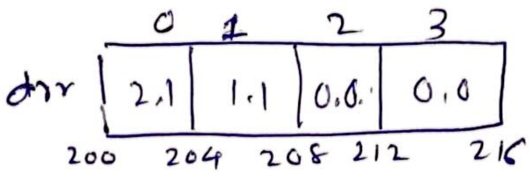
3] int crr[5] = {11, 21};

Crr is a one dimensional array, which has 5 elements, each element is of type integer.



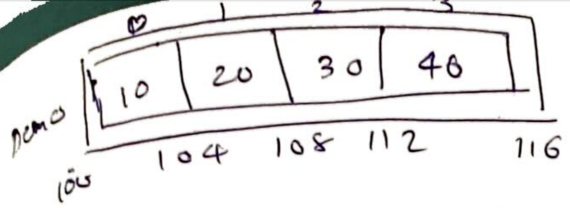
4] float drr[4] = {2.1, 1.1};

drr is a one dimensional array, which has 4 elements, each element is of type float.



5] Const int Demo[4] = {10, 20, 30, 40};

Demo is a one dimensional array, which has 4 elements, each element of type integer & qualifier constant.



Q7] Detect problem in following statement if any.

1] int arr[3] = {12, 23, 45, 65, 87};

(Too many initializers) Compilation error.

The number of initializers in the initialisation list cannot be more than the value of size specifier.

2] int brr[];

In the example, initialisation list is not present, which is mandatory to determine the size of array. In this example the initialisation is not given nor size specification is given.

So, the above declaration is invalid

3] int crr[] = {10, 20, 30}

4] int drr[3+2] = {7+9, 3*2, 78, 9-1};

5] int i = 4;
int arr[i] = {23, 6, 89, 37};

∴) i is a variable & not a constant hence array cannot be initialized.

∴) The size specifier should be compile ~~type~~ time constant expression of integral type.

∴) The memory space to an array is allocated at compile time.

∴) The memory req of array depends upon its element type & the number of elements in it. Hence size of array must be known at compile time so that memory can be allocated.

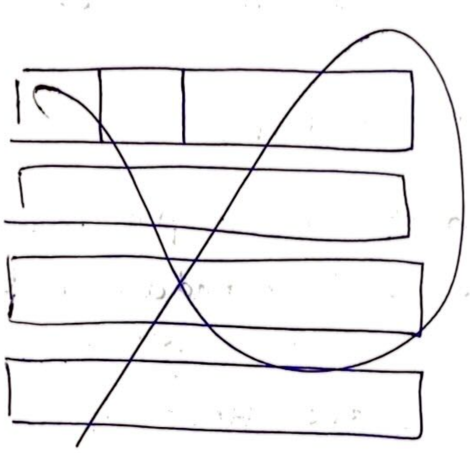
∴) The size of an array cannot be expanded or squeezed at runtime. thus, size must be constant expression so that it cannot be changed at run time.

Q8] What is sizeof operator? Explain with example.

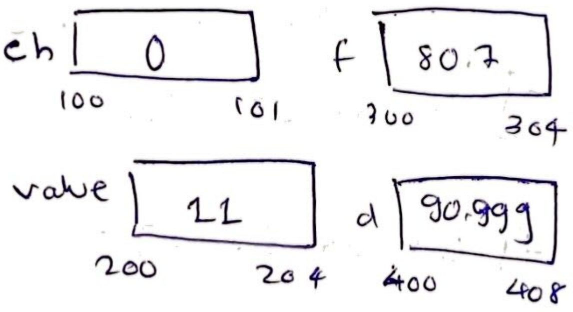
Ans In C & C++ we use the sizeof operator which gives the number of bytes allocated to specific data object.

Eg:

```
int value = 11;
char ch = '0';
double d = 90.999 ;
~~t~~float f = 80.7 ;
```



```
sizeof (value) = 4
size of (d) = 8
size of (ch) = 1
sizeof (f) = 4
```



Q9] Predict the output of below code snippet

```
# ~~ww~~include<stdio.h>
int main()
{
    int arr[3] = { 21, 43, 54 };
    int x = 0;

    x = arr[2] + 21 + arr[0];
    x++;
    printf("%d", x);
  return 0;
}
```

Output

97

Q10] 98.3 , 4.3, 51.6,