

Assignment No-7

PAGE No. / /
DATE / /

- Q) What is meant by array? Explain the use of array with example.

Array

- Array is considered as a linear data structure
- Array is considered as derived datatype in C, C++ & Java
- Array is derived datatype which holds multiple homogenous elements in index format.

Use of Array

- memory for the all elements in Array is Allocated in sequence way
- when we Allocate the memory for Array all the elements are gets sequence memory
- In case of Array we can Access all the elements using Single name So there is no need to multiple names.

Each elements from it Array has its unique memory index.

- In C, C++, Java they Array always start from 0

Syntax: `datatype Arrayname [length of Array] = { data elements }`

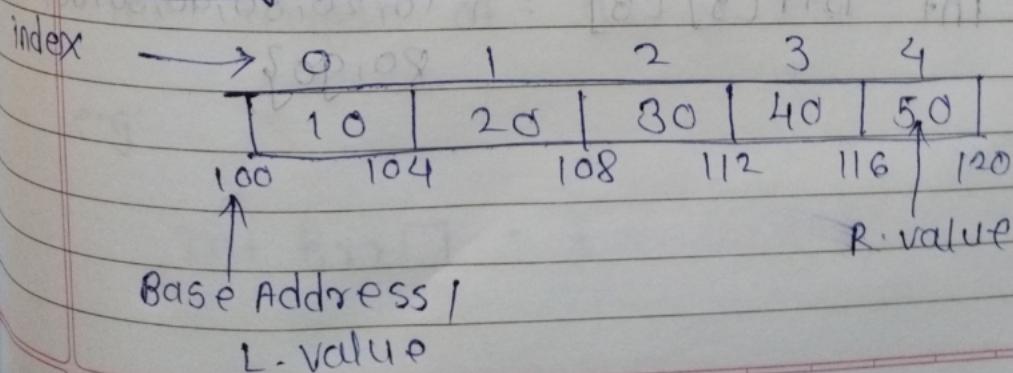
For example:

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

- Array is one dimensional array which contains 5 elements in it.

- each element in it has integer datatype elements initialize with value 10, 20, 30, 40, 50

memory representation : `int Arr[5] = { 10, 20, 30, 40, 50 };`



Q 2] What are the types of array?

→ According to the dimensions there are 3 types of Array.

- 1) one dimensional Array
- 2) two dimensional & multidimensional Array

1) One dimensional Array

e.g.: int Arr [4] = {10, 20, 30, 40};

2) Two dimensional (2D) & multidimensional Array

A multidimensional array is called as Array of Array.

int Brr[8][3] = {10, 20, 30, 40, 50, 60, 70, 80, 90};

Q3) what are the different ways of initializing the elements of array?

following are the different ways of initializing the elements of array

1) With size & initialization list

e.g. $\{10, 20, 30, 40, 50\}$

$\text{int Arr [5]} = \{10, 20, 30, 40, 50\};$

- We can create the Array by specifying the length of array as well as initialize the array immediately using member initialization list.

2) Array initialization with declaration & without length

- At the time of creating Array if we initialized it using member initialization list then it is optional to specify the length of array in []

e.g. :

$\text{int Brr[]} = \{10, 20, 30\};$

3) Declaration & but initialization with less numbers or position

It is also possible to provide the length of array with some value but initialization less no. of elements in it.

e.g.: arr[4] = {10, 20, 30, 40}

4) member by member initialization

e.g.: int arr[3];

arr[0] = 10;

arr[1] = 20;

arr[2] = 30;

5)

(e) what is meant by data structure? and what are the types of data structure?

→ Data Structure

- Data structure is defined as a way of storing & representing data in particular format.
- The concept of data structure is language independent.
- The concept of data structure is applicable for all programming languages.

There are two types of data structure.

1) linear data structure.

2) Non-linear data structure.

Q5) Explain different ways of memory allocation for an Array.
→ following are different types of memory allocation for an Array.

1) static memory Allocation
for static memory Allocation there is no specific function.

- memory is Allocated at compile time in static memory Allocation.
- this type of memory Allocation during compilation has a fixed size.

2) Dynamic memory Allocation

- In the Dynamic memory Allocation memory is Allocated at runtime.
- following are Dynamic memory Allocation functions in C programming

i) malloc :-

- malloc stands for memory Allocation
- In the malloc function the memory is Allocated depend on storage classes of variables.
- the return value of malloc() is void * which indicates address

of allocated memory. ~~mb~~ ~~for~~ [b]
malloc () is accepted for arr [a]

2) malloc ()

- malloc () stands for no calculation.

in this function we pass two parameters

i) no. of elements

ii) size of each element.

3) Realloc ():

- Realloc () is used to allocate & resize the allocated memory size.

Realloc () is used to increase or decrease the size of already allocated memory.

(c) Read the below statement and draw its diagrammatic representation.

- a) int arr[10];
 - b) int brr[2] = {10, 20};
 - c) float crr[6] = {1, 2, 4};
 - d) int drr[5] = {20, 40, 60};
 - e) const int Demo[2] = {10, 20, 30, 40};

a) int arr[10]; () () () ()

- arr is the one dimensional array which contains to ten elements in it, each element has integer datatype.

Diagrammatic layout

index	0	1	2	3	4	5	6	7	8	9
int arr[10]	0	0	0	0	0	0	0	9	0	0
L-value	190	104	108	112	116	120	124	128	132	136

- b) `int brr[2] = {10, 20};`

Diagrammatic representation

brr [10] 10 20 30 40 50

longint 100 104 108 112 116 120

float brr [5] { 1.1, 2.4, 3.3 }

brr is the one dimensional array

we declare with 5 elements but

initializing 3 elements having datatype float.

Diagrammatic representation

index → 0 1 2 3 4 5

crr [100] 100 104 108 112 116 120

L-value R-value

crr [5] { 20, 40, 60 }

drr is the one dimensional array
declare with 15 elements but initializing
3 elements having datatype integer.

index → 0 1 2 3 4

drr [100] 100 104 108 112 116 120

L-value R-value

c) constant Demo [2] = {10, 20, 30, 40};

- this array is invalid, because if we when we create any constant array then we cannot increase decrease the initializing element of that array.
- In this array Demo having declare with 2 elements but initializing with 4 elements this is not allowed because array Demo has constant.

Q1] Find out the problems in the following statements if any & correct it.

a) int arr [4] = {1, 2, 3, 4, 5, 6, 7};

- in this array declaration with 4 elements & initializing with 7 elements this not allowed in array declaration & initialization.

- we can initialize less elements but not declare the declaration with less elements around 2nd mistake & correction

int arr [4] = {1, 2, 3, 4};

OR

int arr [4] = {1, 2, 3, 4};

b) `int brr[];`

- in this Array `brr` the size is not mentioned/declared & initialized correction:-

~~(S.P.A. 2018-19) int brr [] = { 10, 20, 30, 40 };~~
~~int brr [2] ;~~

~~[] int crr [] = { 10, 20, 30, 40, 50 } ;~~

- In this Array no. of any problem

~~of d) int drr [3+2] = { 7+1, 3*5, 100, 10-13 } ;~~

- this is also valid but we can initialize elements with members initializing by listing & the size of array is

~~int drr [3+2] = 5~~

~~drr [0] = 8~~

~~drr [1] = 15~~

~~drr [2] = 100 ;~~

~~drr [3] = 9 ;~~

~~by default drr [4] is 0~~

e) `int i = 4;`

`int arr[i] = {2, 3, 4, 5, 23, 45, 67, 89};`

Ans: Array cannot initialize using variables.

Correction

`int arr[4] = {23, 45, 67, 89};`

Q8) What is mean by sizeof operator explain with examples.

In C & C++ language we use the sizeof operator which gives the no. of bytes allocated to the specific data object.

The sizeof() operator returns an unsigned integer value which represents the amount of memory in bytes occupied by the operand.

By using the sizeof operator we can find the length of Array.

e.g.:

`int length_of_Array = sizeof(Array);`

`or
length_of_Array = sizeof(Array[2]);`

Syntax:-

PAGE No.

DATE

/ /

so if you want to know
size of type;

e.g:-

```
#include <stdio.h> // [1]
int main () {
    int a = 5;
    printf ("Size in bytes :- %u",
            sizeof (a)); // [1]
```

O/P:-

Size in bytes : 4 // [1]

Eg) Predict the output of below code snippet

```
#include <stdio.h> // [1]
int main () { // [1]
```

int arr [3] = { 21, 43, 50 };

int x = 0;

x = arr [2] + 2 * arr [0];

x++;

printf ("%d", x);

return 0;

}

output:-

output of above code is 82

Q 10) # include <stdio.h>
 int main () {
 }

```
float arr[4] = {98.3, 4.5, 51.5, 7.5};  

int i = 0; // Initialize  

printf("%.f :: ", arr[i]);  

i++;  

printf("%.f :: ", arr[i]);  

i++;  

printf("%.f :: ", arr[i]);  

return 0;
```

Ans To explain std library (P)

O/P:-

98.300000 : 4.500000 : 51.500000

() nimm hui

g

{ 98.300000 : 4.500000 : 51.500000 }

: 0 = x hui

: 0 = x + [51.500000] = x

; H-X

, (x, "hui") thing

{ o nimm }

}