

Arrays

- Why are we grouped into classes?
- Why are we grouped anywhere for that matter? And how does that grouping occur?

Why arrays?

- Handling one is better than handling many variables
- Grouping of similar data
- Easy manipulation

Array syntax

- Can be single dimensional or multi dimensional

Eg:

`arr1[30]` //one dimensional array of size 30

`arr2[5][6]` //two dimensional array of size 5*6

`arr3[4][5][6]` //three dimensional array

and so on..

More about arrays

```
int a[5]
```

Means:

a is an integer array of 5 elements

a is array name

int is the data type for the array

IMP: array index starts from 0

Also there is no bounds checking in C

a[3] is not the 3rd element but 4th

Array

- Nothing but a pointer
- Array elements are stored in contiguous memory locations
- A pointer when incremented always point to an immediately next location of its type
- Accessing array elements by pointers is faster than compared to subscripts
- `num[i]`, `i[num]`, `*(num+i)`, `*(i+num)` all are same

2-D arrays

- The first subscript is row number and second is column number
- Row number is optional column number is compulsory
- $s[2][1] = *(s[2] + 1) = *(*(s+2) + 1)$

Summary

- An array is similar to an ordinary variable except that it can store multiple elements of similar type.
- Compiler doesn't perform bounds checking on an array.
- The array variable acts as a pointer to the zeroth element of the array. In a 1-D array, zeroth element is a single value, whereas, in a 2-D array this element is a 1-D array.
- On incrementing a pointer it points to the next location of its type.
- Array elements are stored in contiguous memory locations and so they can be accessed using pointers.

Questions

- Write this in words `int arr[3][4][5]`
- Output:

```
static int arr[]={0,1,2,3,4};  
int *p[]={arr,arr+1,arr+2,arr+3,arr+4};  
int **ptr=p;  
ptr++;  
print(ptr-p,*ptr-arr,**ptr);  
*ptr++; //print same above  
*++ptr; //print same above  
++*ptr; //print same above
```

- if array begins at 1898320 what is the output

```
int arr[]={1,2,3,4,5,6}
```

```
print(arr,&arr,arr+1,&arr+1)
```

- Are arr and &arr same for an array of 10 integers

```
int a[2][2] = {1,2,3,4};
int i,j;
int *p[]={(int *)a, (int *)a+1, (int *)a+2};
for (i=0;i<2;i++)
{
    for (j=0;j<2;j++)
        print(
            *(*p+i)+j),
            *(*p+j)+i),
            *(*i+p)+j),
            *(*p+j)+i))
}
```

Answers

- Arr is collection of 3 2-D arrays each containing 4 rows and 5 columns of integers
- 111 222 333 344
- 1898320 1898320 1898324 1898340
- Nope
- 1111 2222 2222 3333

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com