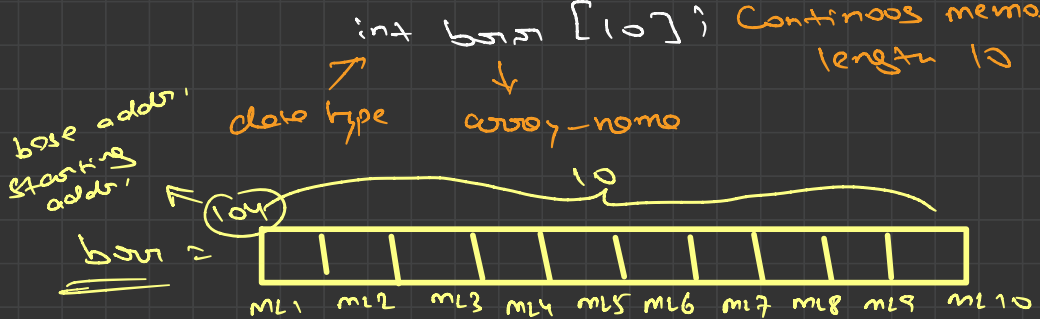
array basics 1

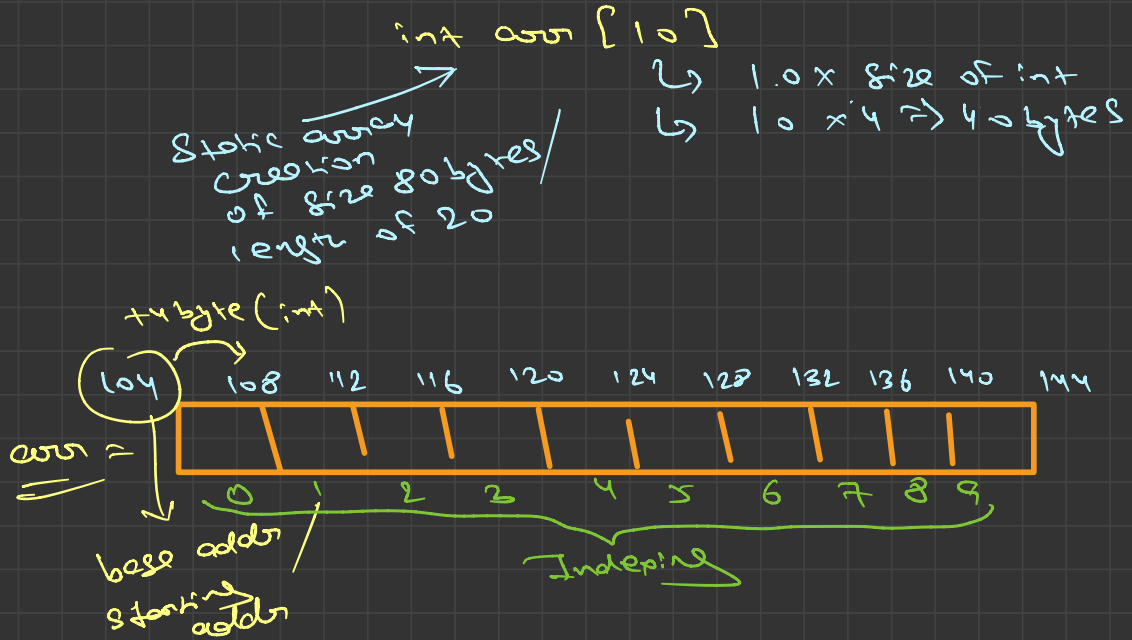
What is array ?

Some type data structure

- list of similar items
- collection of elements
- data structure
- continous memory location



Calculate space ?



Creation of array

int => int arr[127];

Char => Char alpha[26];

bool => bool flags[27];

long => long num[10];

Short => Short num[15];

How to check from which addr our array start?

Address operator => & variable name

```
int arr[10];

cout << &arr;
cout << arr;
```
} some output

Size of array

```
int arr[10];
```

Size => sizeof(arr) => 40 bytes

length => sizeof(arr) / sizeof(int) => 10

Array initialization

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

```
int brr[5] = {1,2,3,4,5};
```
=> | 1 | 2 | 3 | 4 | 5 |

```
int crr[5] = {1,2};
```
=> | 1 | 2 | 0 | 0 | 0 |

```
int drr[2] = {1,2,3,4};
```
=> error

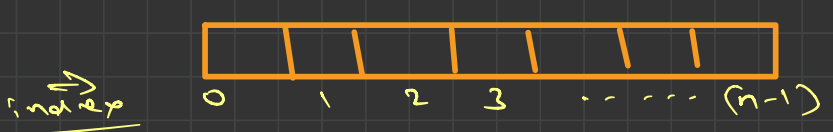
Bad practice

```
int n;
cin >> n;
int arr[n];
```

if we don't have the continous memory avoible the user won't it throw the error

Indexing in array

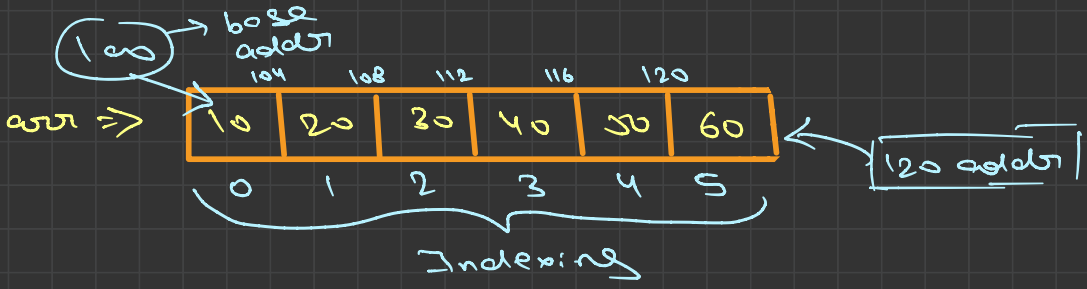
array => n size



Starting from 1 based numbering
0 based indexing

address in array → to access the array

int arr[ ] = {10,20,30,40,50,60}



formula

arr[i] => value at (Base addr + (data type * index))

arr[0] = 10 ? 1400 ?

int = 4 byte

arr + indexing * size of int

100 + 0 x 4 => 100 addr

arr[1] = 20

100 + 1 x 4 = 104 addr

arr[5] = 60

100 + 5 x 4 = 120 addr

Traversing an array

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

for (int i = 0 ; i < n ; i++)
{
    cout << arr[i] << endl;
}
```

Input into array

```
int arr[5] ;
int n = 5;

for ( int i = 0 ; i < n ; i++)
{
    cin >> arr[i] ;
}
```

Array & functions

```
int main()
{
    int size = 5;
    int arr[size];
    Solve(arr, size);
}
```

```
Solve(int arr[], size)
{
    ;;;;
}
```

Dynamic memory allocation

```
int arr[5];
```

Static array → Static memory allocation

```
int *arr = new int[n];
```

↳ dynamic memory allocation (heap)

↳ change the size of array at runtime

if we dont assign the values, it will have garbage or 0 by itself

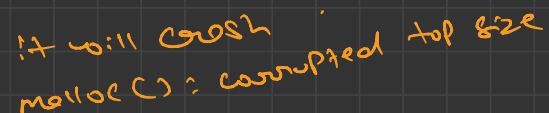
```
int n;
cin >> n;

int *arr = new int[n]; // each ele → 0 or garbage

// taking n elements inputs & inserting in array

for (int i=0 ; i<n ; i++)
{
    int data;
    cin >> data;
    arr[i] = data;
}// →  0 → 4 index

// let me try to insert more items
// 5 → 14 index
for (int i=0 ; i<10 ; i++)
{
    arr[n+i] = 80;
}
```



To overcome this → Vector used