Declaration & Initialization of an Array :

unknown elements

✓ = new int[size]
✓ = new int[size][size]

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

[1, 10, 20, 30]

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$$

3 x 3

Alternative ways of declaring Array :

1D Array

```
→ int ar[]
  int []ar
  int[] ar
```

2D Array

```
int ar[][]
int [][]ar
int[][] ar
int []ar[]
int[] ar[]
int[] []ar
```

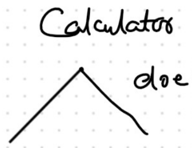
MCQ

{ ? ≡ }

Var args [Variable no.of arguments] !

```
int add( int... a ) {
    ___
    ___
    ___
}
```

c



range

2+2
2+2+2
3+2+1+2

1,2,3,4

1+1 →
2+1 →
3+1

add( )
a,b

```
for(          ) {
  (1,2)
  (res , 3)
  (res, 4)
}
```

Note :

→ In above program add(), can accept any no.of arguments of same data type

→ The arguments which it collects is stored in the array format

→ The size of the array will be estimated by compiler & it counts the no.of arguments passed in method call

→ The elements passed can be accessed through the index values

Alternative ways

```
int add(int... args){ ... }
int add(int ...args) { ... }
int add( int ... args) { ... }
```

Rules & regulations

[a, b] [... args]

① We can have only one var args in a method

```
int add( int...args1 , double... args2) { ... }  ✗
```

② In case if we are having var args along with normal parameters the var args should be always declared at the last

```
→ add( int x , int ... args){ ... }

add(1) ✓
add(1,2)        x=1
                args1 =[2]

add(1,2,3)      x=1
                args = [2,3]

add( ) ✗
```

③ If we have two methods were one method accept normal parameters & other method accepts var args elements then priority/preference is given to normal parameters

```
①  int add(int x ,int y){          |   Calc.add()          ②
        ≡                          |   Calc.add(10)        ②
    }                              |   Calc.ad(10, 20)     ①
②  int add(int... args){           |
                  ↳ 0,1,2 ....n    |
        ≡
    }
```

④ In main method var args can be create with '[ ]' (or) "..." both are allowed, but when try to use ver args in usr defined method '[ ]' → as a parameters which accepts array not as var args

```
. add (int a[ ])
add([1,2])
```

[5] {1,2,3,4,5} → size is fixed

Array Disadvantages

1. We can only store Homogeneous type of data in an array

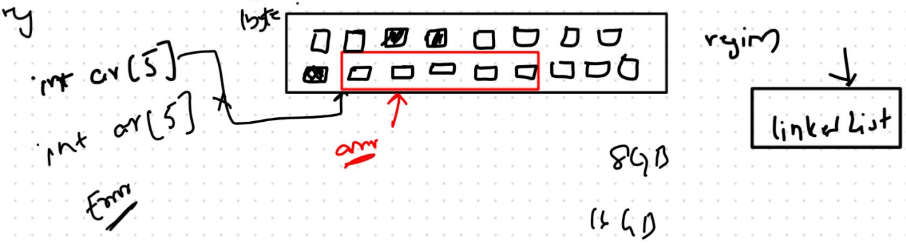
2. The size of an array is fixed because we cannot increase (or) decrease after array is defined

[5] [6]

1024 → 1GB → 1024MB → Bytes
→ Bits

chunks

3. It needs contiguous memory



16GB

32GB

Advantages ✓

1. Elements of array can be accessed through index