# Arrays in Java

Selvamani Kannan

## Agenda

- Recap recursion, Divide and Conquer
- Primitive data types
- Arrays
  - Need for an array and examples
  - o 1D, 2D
  - Implementation
- Problem solving

### Recap

- Recursion Recall
  - recursion
  - Base case
  - Recursion tree
  - Stack space
- Divide and conquer

#### Primitive data types

#	Data types	Size
1	int	32 bit
2	float	32 bit
3	char	16 bit
4	double	64 bit
5	byte	8 bit
6	short	16 bit
7	boolean	1 bit
8	long	64 bit

# Arrays

## Array

- Int a = 1
- int b = 2
- int c = 3
- int d = 4





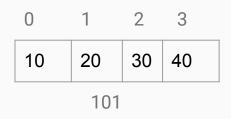
### Array

- Finite length
- All elements are of same type
- Index based (0 as starting index)
- Cannot increase

- Declaration
- Instantiation
- Initialization

#### Arrays - Heap memory

```
int arr[] = new int[4];
System.out.println(arr[0]);
arr[0] = 10;
arr[1] = 20;
arr[2] = 30;
arr[3] = 40;
arr[4] = 50; // Error
int nums[] = {11,21,31,41};
```



Heap Memory		
variable	Address	
arr	101	

#### 2D array

```
int a[] = {2,4,6,8};
int b[] = {1,3,5,7};
int c[] = {9,1,2,3};

int arr[][] = {a,b,c};
int arr[][] = {{2,4,6,8}, {1,3,5,7}, {9,1,2,3}};
```

0	1	2	3
---	---	---	---

0	2	4	6	8
1	1	3	5	-
2	9	1	2	,

- Jagged array

#### Printing array

```
int arr[][] = {
int arr [] = {2,4,6,8};
                                                             {2,4,6,8},
                                                             {1,3,5,7},
                                                             {9,1,2,3}
     for(int i=0;i<arr.length;i++){
            System.out.println(arr[i]);
                                                             for(int i=0;i<arr.length;i++){</pre>
                                                                   for(int j=0;j<arr[i].length;j++){
                                                                          System.out.println(arr[i][i]);
     for(int k: arr){
                                                             for(int k[]: arr){
    for(int i:k){
            System.out.println(k);
                                                                         System.out.println(i);
```

#### Complexity

Operation	Complexity	
Insert, Delete	Need to create an new array and copy O(n)	
Access, Update,	Using index, directly access or set value O(1)	
Size	Fixed	
Memory allocation	Contiguous	

## Questions

## **Problems**

#### Reverse an array

Reverse the given array

Sample Input and Output:

- 1. Input: {1,2,3,4,5} Output: {5,4,3,2,1}
- 2. Input: {10,51,93,77,61} Output: {61,77,93,51,10}
- 3. Input: {'N', 'e', 'w', 't', 'o', 'n', ' ', 'S', 'c', 'h', 'o', 'o', 'l'}
  Output: {'l', 'o', 'o', 'h', 'c', 'S', ' ', 'n', 'o', 't', 'w', 'e', 'N'}

# Thank you!!!