LINKED LIST { Easiest DS }

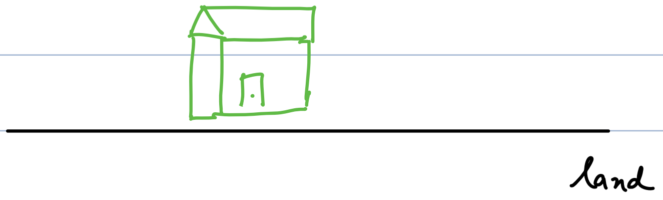
Classes & Objects

Linked list

Question in WA group tomorrow

Try those questions

Class ?



Architect [Blueprint]

Class is just a blue print it does not occupy any space

```
class Employee {
    String name
    long salary
    Strirg gender
    String designation
}
```

Employee emp = new Employee()

emp — object reference

new Employee() — object

emp ⤵ emp2 ⤵

name:	Deepak
salary:	~~1000000~~ 10
gender:	
desig:	

Objects

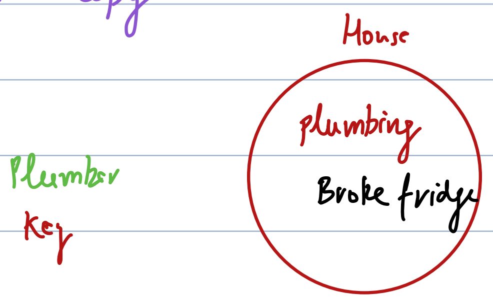
Instances of the class

emp. name = "Deepak"

emp. salary = 10000000

emp2 = emp // Shallow copy

emp2. salary = 10



1 object ← multiple object referecu

emp2 = new Employee()

emp

name: Deepak
salary: 1000000
gender:
desig:

emp2

name: Karan
salary:
gender:
desig:

emp2.name = "Karan"

~~emp = emp2~~

~~print(emp.name)~~

↳ ~~Karan~~

emp3 = new Employee()

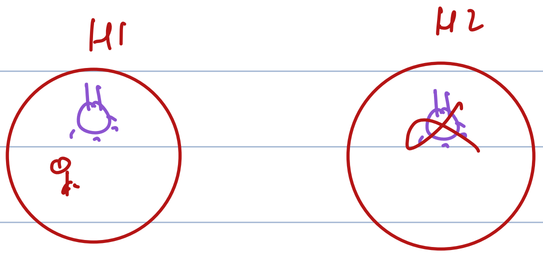
~~emp3~~ →

name: Nikhil
salary:
gender:
desig:

emp3.name = emp2.name

emp3.name = Nikil

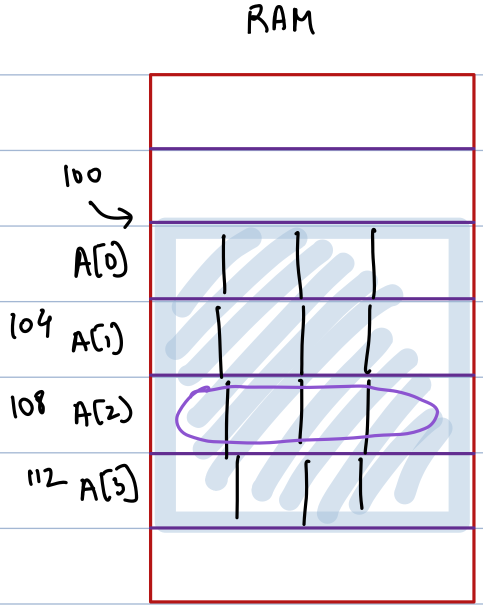
print (emp2.name)



Object refernces → memory address of the object

Arrays

$A[i] \longrightarrow O(1)$



Array : Contineous memory allocation

int A[4]

int → 4 bytes

A[2]

↓

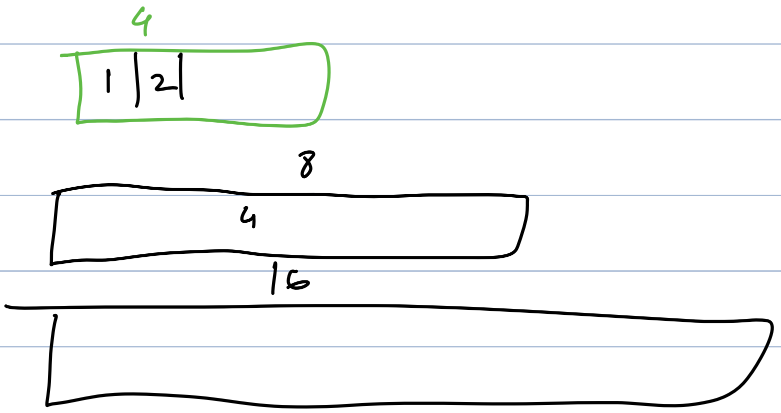
$100 + 2 \times 4 = 108$

A[1]

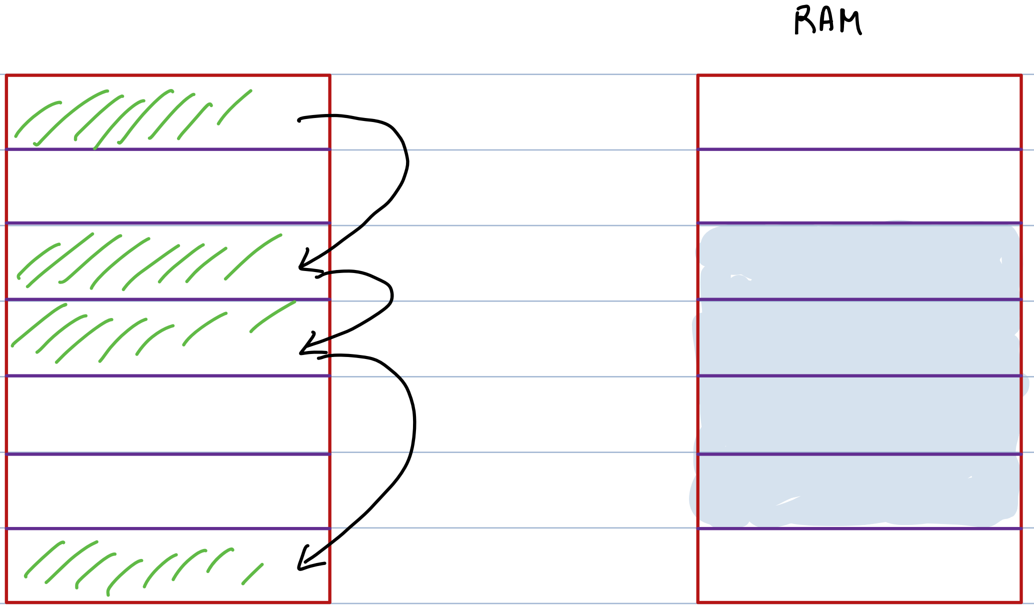
↓

$100 + 1 \times 4 = 104$

Dynamic Array

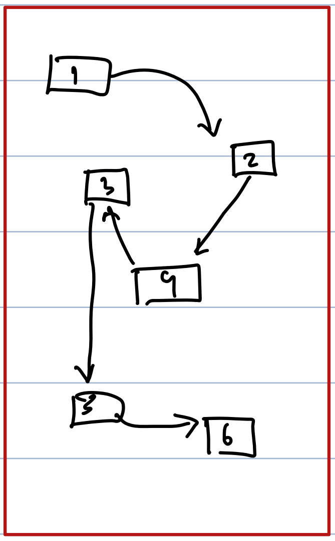


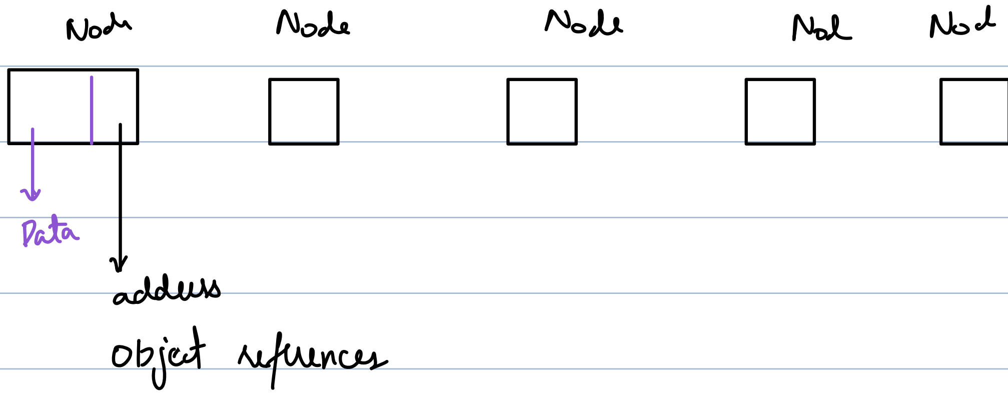
Arrays



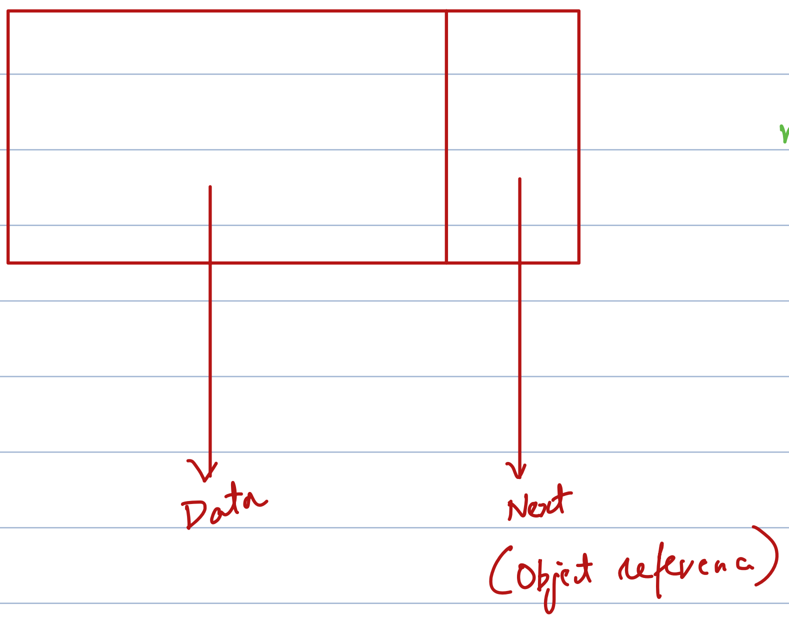
Linked list

Memory allocation is not contineous

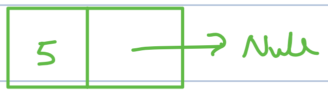




```
class Node {
    int data;
    Node next;
}
```



n = new Node()



n.data=5

n.next = null

m = new Node()

m.data = 3

m. next = NULL

Constructor

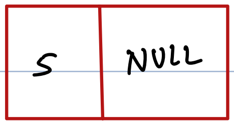
Used to initialize objects

→ Method / function with no return type

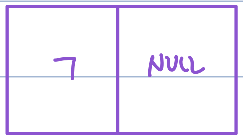
→ Name of function should be same as class

```
class Node {
    int data;
    Node next;
    Node (int  x) {
        data = x
        next = null
    }
}
```

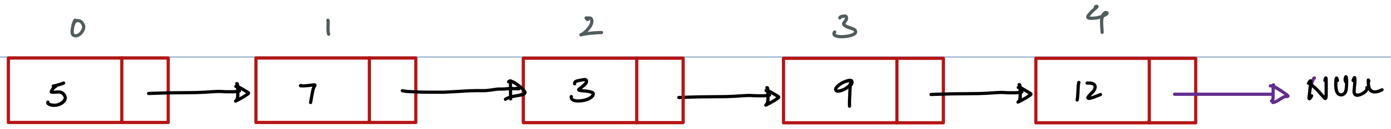
Node n = new Node(5)



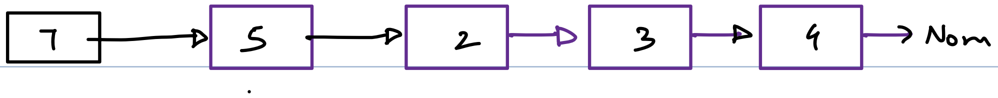
Node m = new Node(7)



Q1) Print a linked list

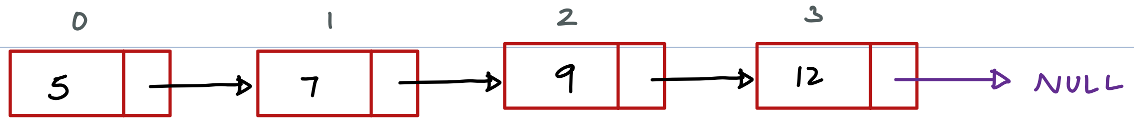


Insert at first



k will be valid value

Q2) Insert at k^{th} position in LL $0 \le k \le N-1$



}