LINKED LIST { Easiest DS }

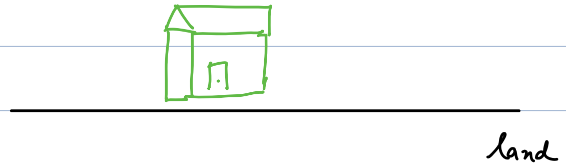
Classes & Objects

Linked list

Question in WA group tomorrow

Try those questions

Class ?



Architect [Bluepeint]

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

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

Employee emp = new Employee()

object reference object

emp. name = " Deepak "

emp. salary = 10000000

emp ↴ emp2 ↴

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

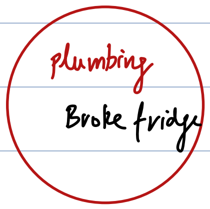
Objects

Instances of the class

emp2 = emp // Shallow copy

emp2. salary = 10

House



Plumber

Key

1 object ← multiple object refereu

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()

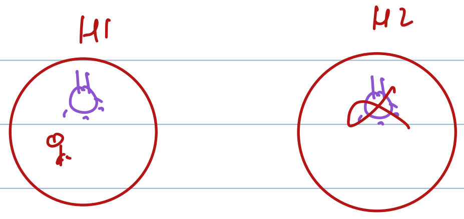
~~emp3~~ →

name: Nikhil
salary:
gender:
desig:

emp3. name = emp2. name

emp3. name = Nikil

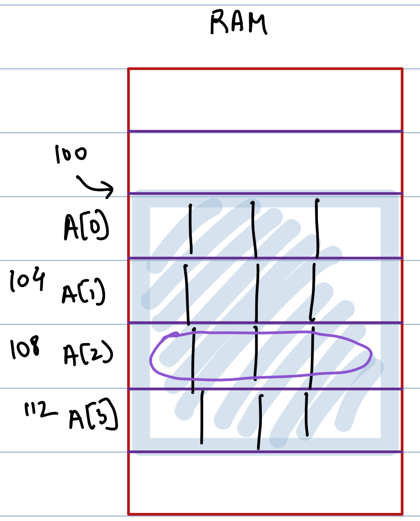
print (emp2. name)



Object refernces → memory address of the object

Arrays

A[i] → O(1)



Array : Contineous memory allocation

int A[4]

int → 4 bytes

A[2]

↓

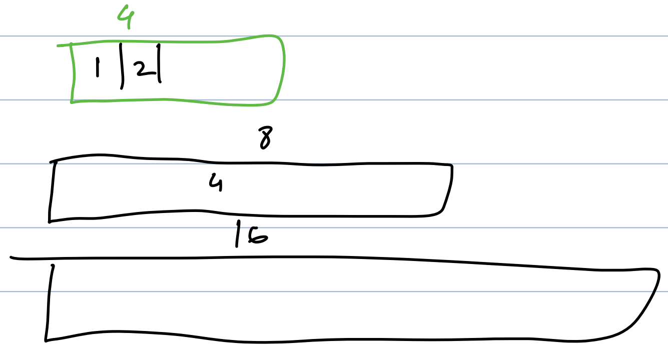
$100 + 2 \times 4 = 108$

A[1]

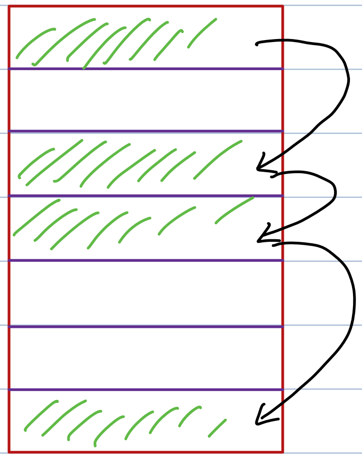
↓

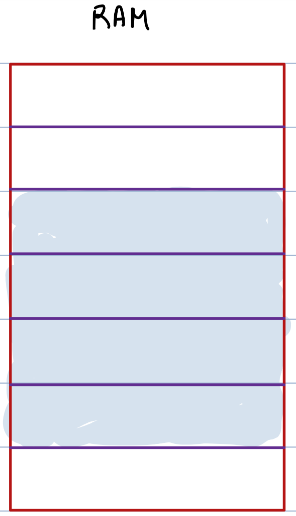
$100 + 1 \times 4 = 104$

Dynamic Array



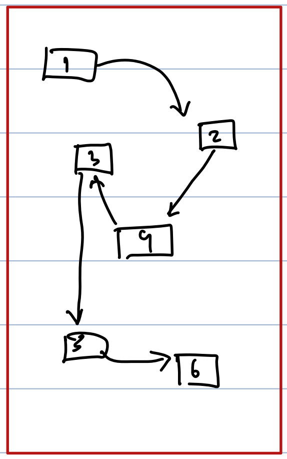
Arrays

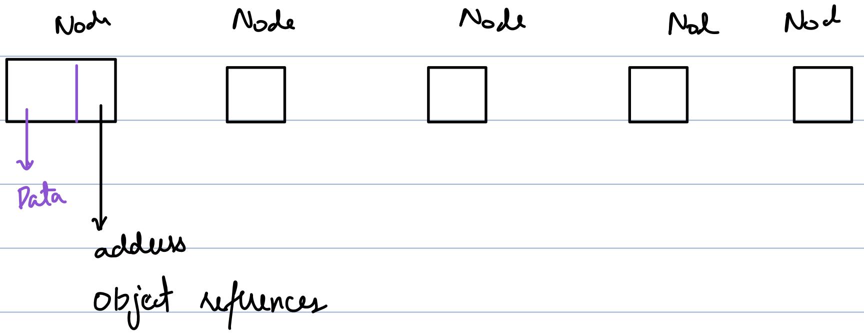




Linked list

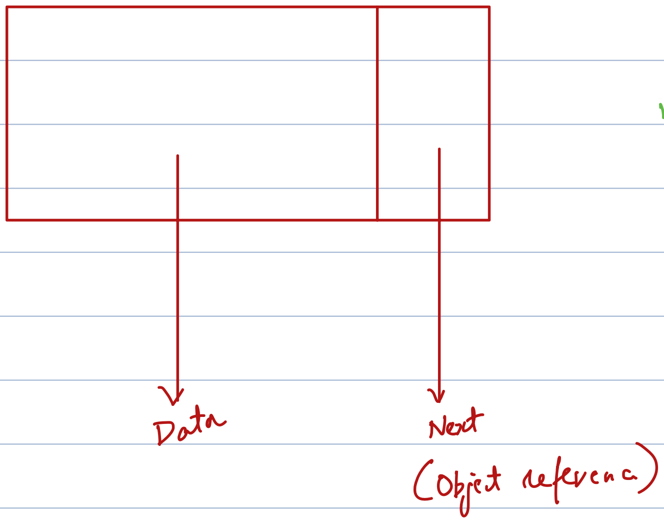
Memory allocation is not contineous

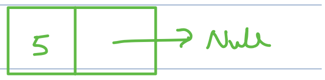




```
class Node {
    int data;
    Node next;
}
```

n = new Node()





n.data=5

n.next = null

m = new Node()

m.data = 3

m. next = NULL

Constructor

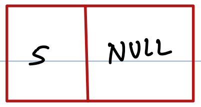
Used to initialize objects

→ Method / function with no return type

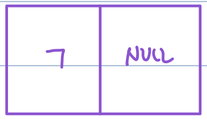
→ Name of function should be same as class

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

Node n = new Node(5)



Node m = new Node(7)

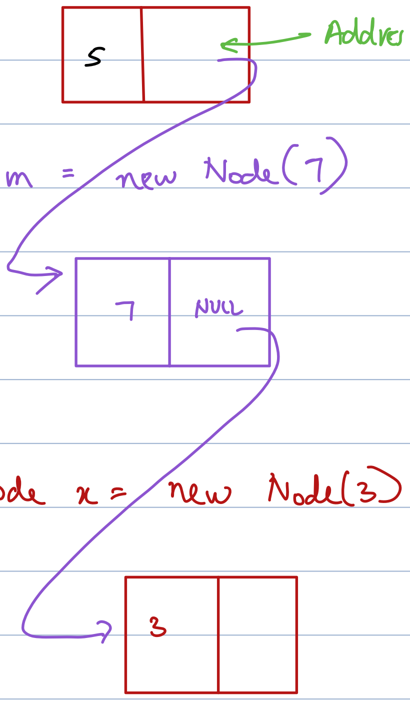


10:28 - 10:43

n.~~data~~ → 5

n.next → null

Node n = new Node(5)



Addres of node 7

n.next = m

Node m = new Node(7)

n.next.next = x

Node x = new Node(3)

n.next.data → 7

n.next.next.next → null

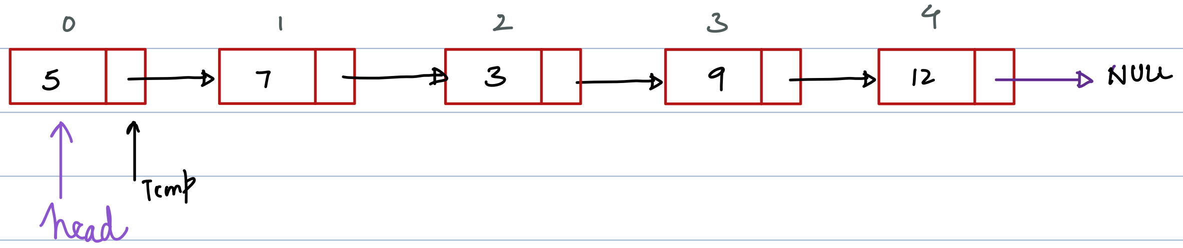
n.next.next.next.data → Null Pointer Exception

NULL . NPE

1) Can only go forward in a LL

2) First node of every LL is head

Q1) Print a linked list   temp.data = 9

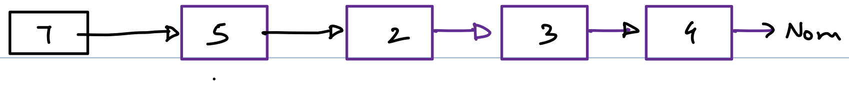


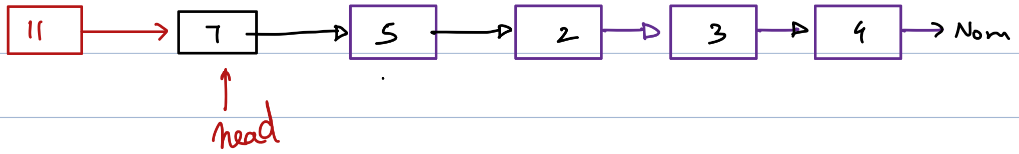
Referena to the head nod

```
void printLL(head){
    temp = head            c=0
    while (temp != NULL)
        ~~print(temp.data)~~ → C++
        temp = temp.next
    }
}
```

Size of a LL

Insert at first

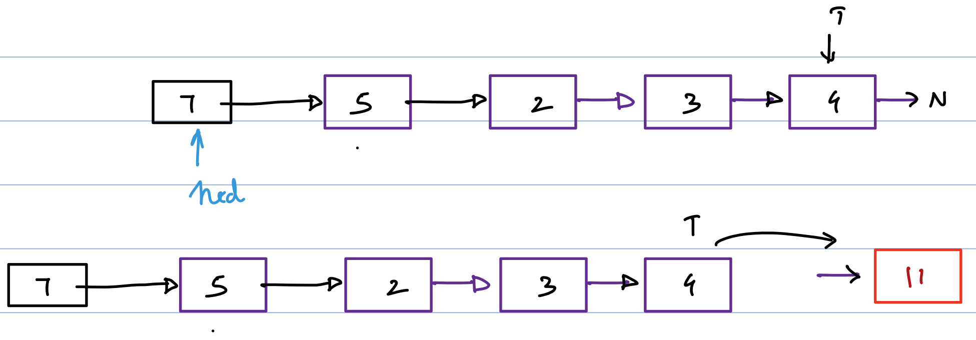




nn

```
Node insertAtFront( head, data){
    Node nn = new Node(data)
    nn.next = head
    head = nn
    return head
}
```

Insert at end



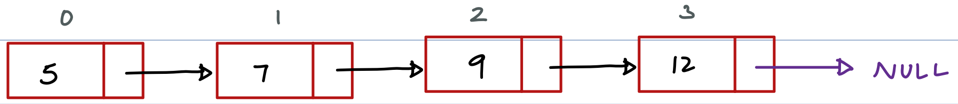
```
Node insertAtEnd ( Node head , int data ){
    Node nn = new Node(data)
    if (head == null) { return nn }
    Node temp = head
    while ( temp.next != null) {
        temp = temp.next
    }

    temp.next = nn
    return head
}
```

TC : O(N)

Done!

k will be valid value

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


0
1
2
3
5
7
9
12
NULL