Array → Size fix

↓

collection of

[Similar DT

↓ Objects → classes

→ Primitive Data → byte

short

int = 10

long = 23.5

float, double

char, boolean

↓

Single value

Collection → Objects

↓

Multitype DT values:

int[ ] arr = new int[5];

remove → array size

↑ int

[10, 45.7, True, 'a', "str"]

→ No fixed size

changable in nature

N.P → Integer class

↓

collection → concept/framework

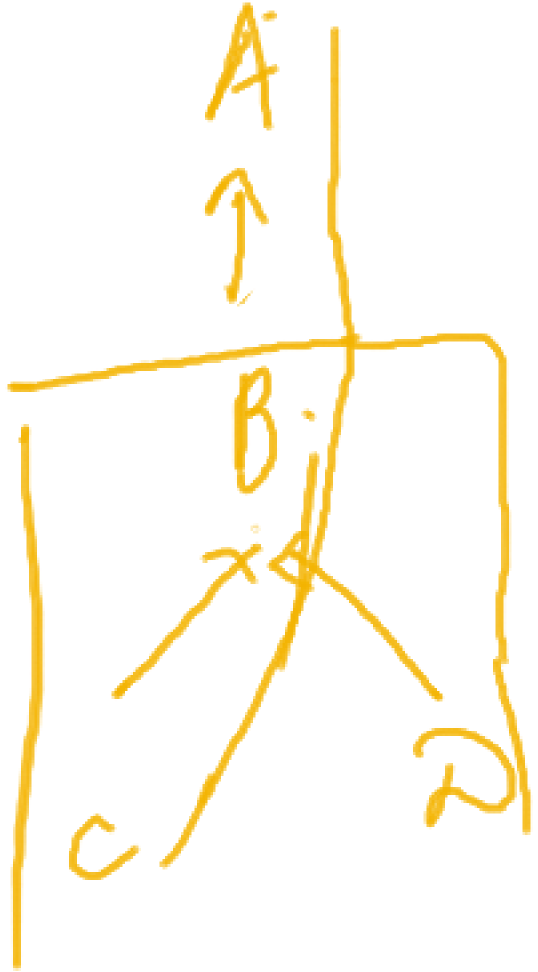
Collection → Interface

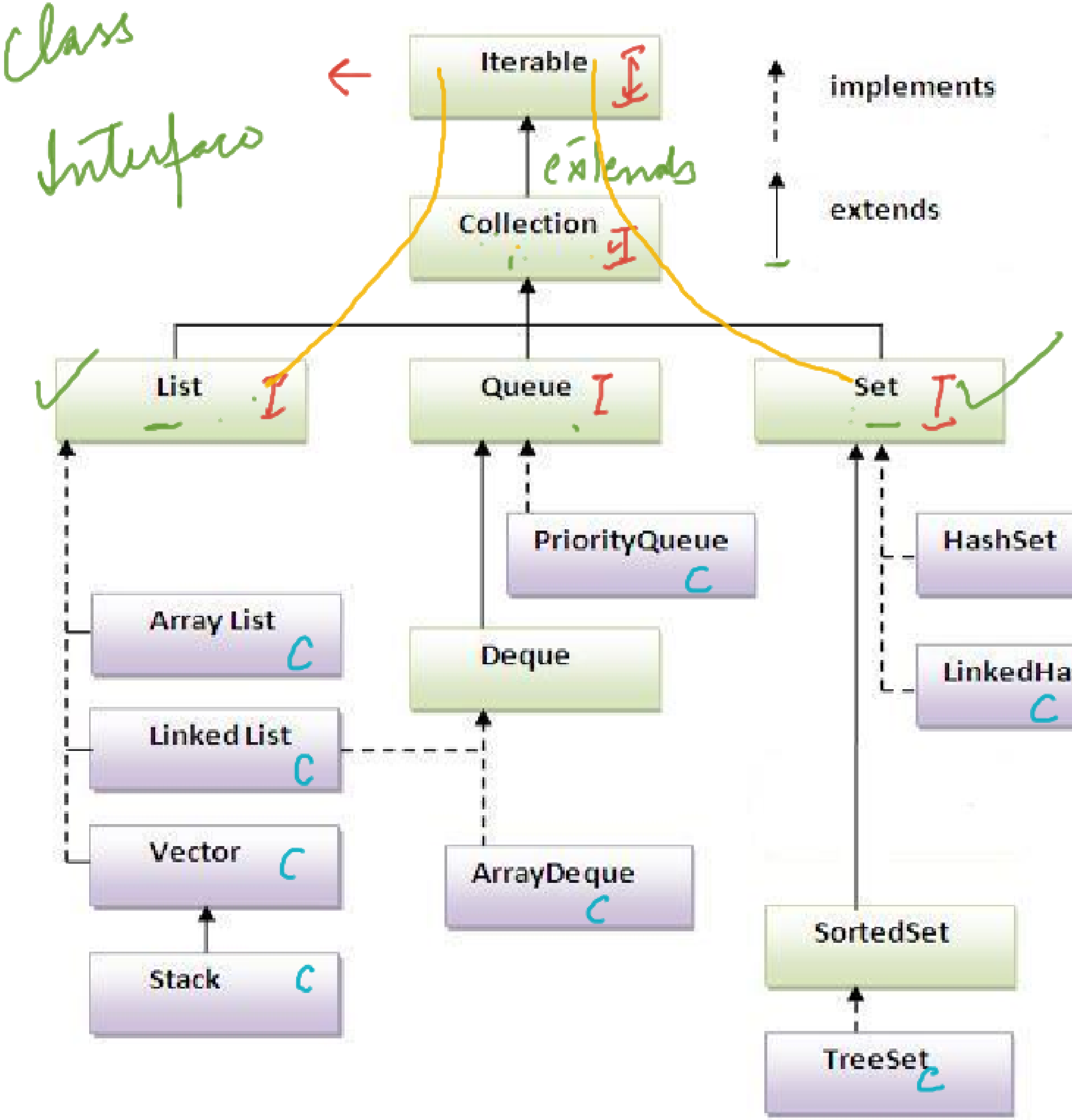
Collections → Class

Throwable → Class

Iterable → Interface

↓




Iterable

Collection

List

Queue

Set

PriorityQueue

Deque

Array List

Linked List

Vector

Stack

ArrayDeque

HashSet

LinkedHashSet

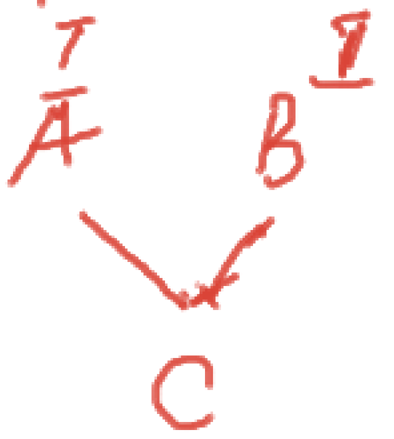
SortedSet

TreeSet

implements

extends

Multiple Inheritan



C extend G

List :- (i) Ordered in nature

(2) Duplicate elements can be stored

100 elements → 1
2
3
4
5
6
7
8
8
5
7

List
↓
I
↓
Parent → Collection
↓ Parent
Iterable

```
interface I1 {
    public void m1();
                m2();



}
```

1.8

→ default

→ static

type Integer

Collection → Can Store objects

object

obj1 obj2 obj3

{10, 20, 30} int X

10 → 1 byte - 8 bits

→ Wrapper classes

→ P.D

→ N.P.D

→ Wrapper classes

P.D.T

byte → Byte

short → Short

int - Integer

long - Long

float → Float

double - Double

Byte → Wrapper classes

P.D → classes

Char - Character

boolean - Boolean

add
add All
remove
get
size

Non Static

(I) Collections → Methods declare

No Def.

add()
remov

(I) List

ArrayList class: implements List

```
public void add( ){

}
```

ListInterface

add();

=

Class Arraylist implements List

```
public void add(){

}
```

Custom class

List. add(); ✗

```
Arraylist al =
        new Arraylist()
```

al. add(10);

method → overridden

method prop.

=

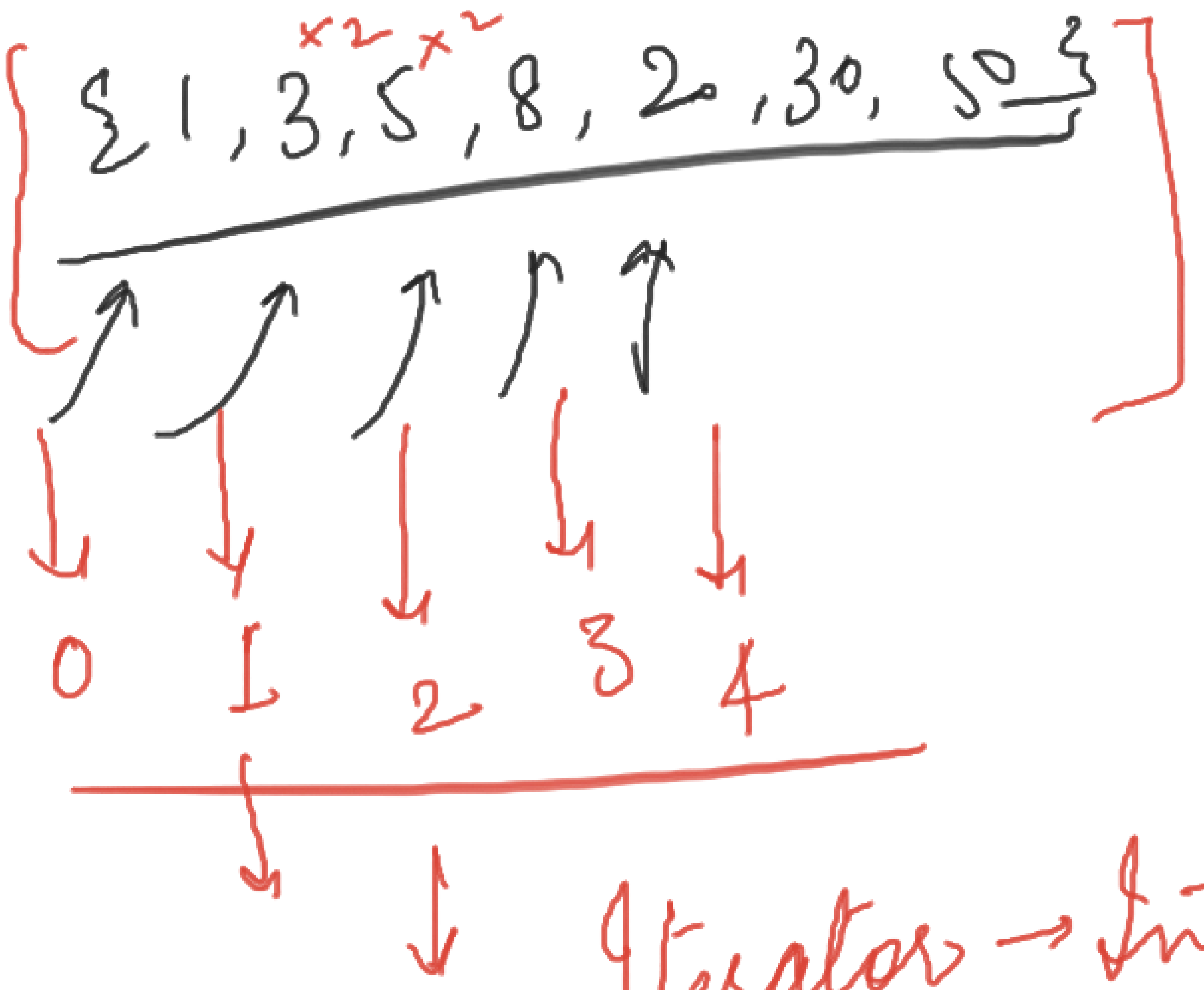
Parent ref var = new Child class Name();

Class

interface

✓ List l = new Arraylist();

Array →



→ Iterate

Iterator → Interface

1 ×2
3 ×2
5
8
20
30
50

for

— for each ✓

✓
```
for ( init; cond; inc/dec ) {

}
```

iterator

↓

```
for ( int x as arrName ) {

}
```

Iterator → Interface

↓

hasNext() →

next()

remove() →

Linked List :

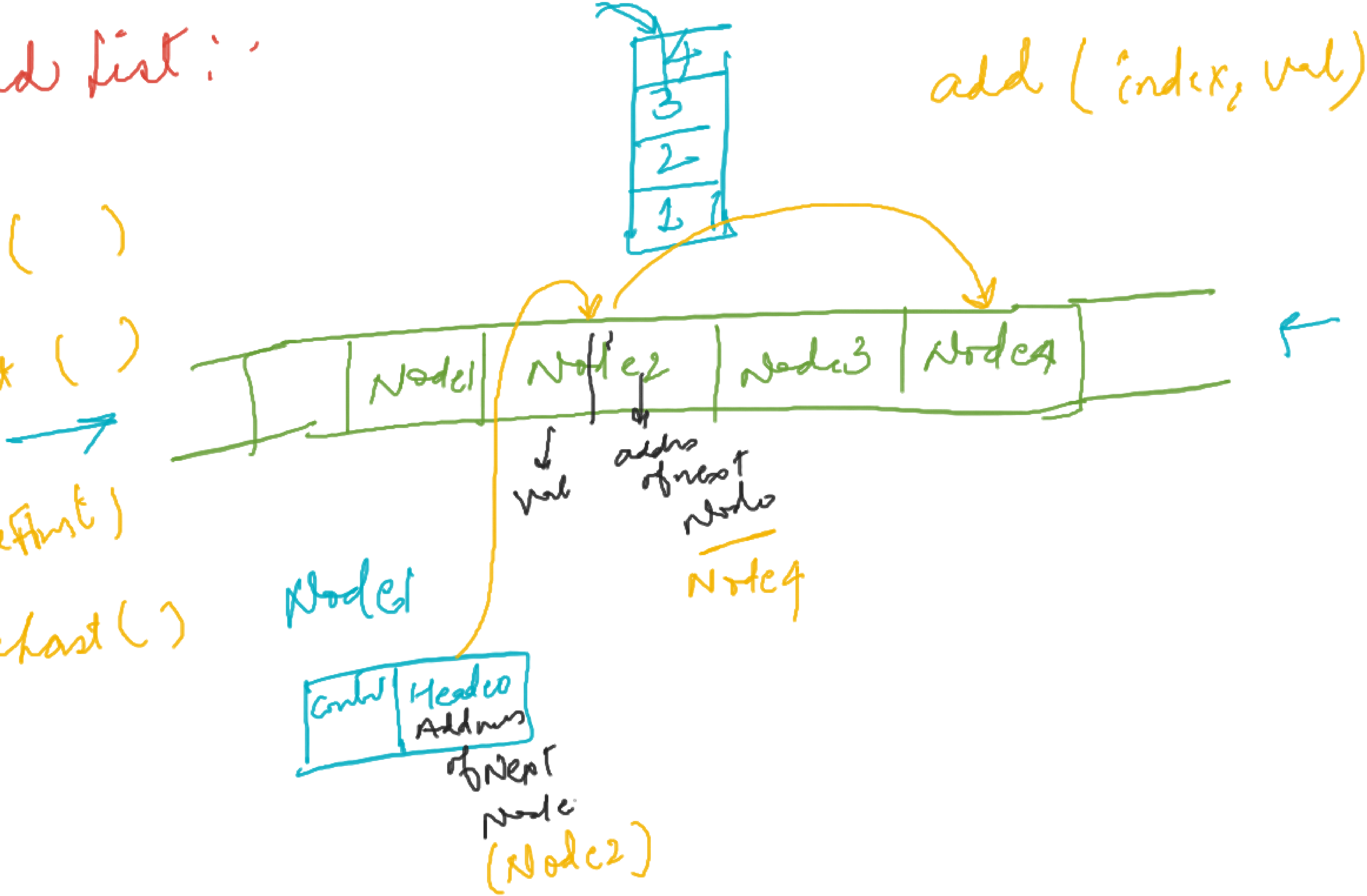
add (index, val)

addFirst ( )

addLast ( )

removeFirst( )

removeLast( )



4
3
2
1
Node1
Node2
Node3
Node4
val
addrs of next Node
Node4
Node1
Content
Header Address
of Next Node
(Node2)