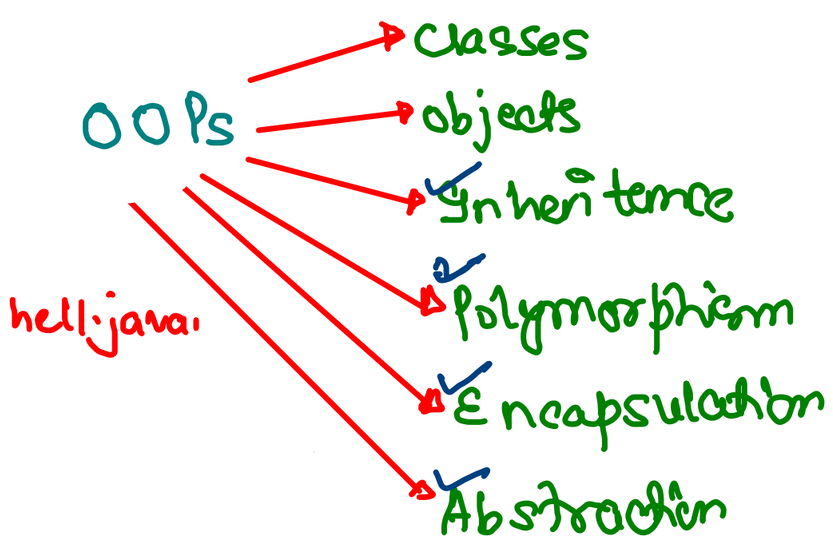


Polymorphism :

① Run time polymorphism

② Compile time polymorphism.

eg - overloading overriding

NOTE: operator overloading is Not allowed in JAVA

Inheritance.

① Super class

② Sub class

③ Reusability.

Encapsulation: Wrapping of Data Members and function together in a class in called Encapsulation.

Abstraction: Only essential Information will display to the user.

Access Modifier ⟶ public
private.

Class: A class is a user defined blue print or prototype from which object are created. It represent the set of properties or methods that are comon to all objects of one type.

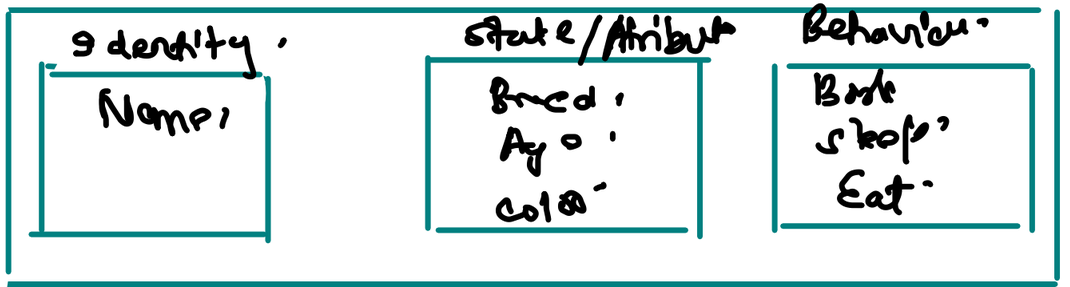
Component of class:

1. Modifiers.
2. class Name.
3. super class (if any)
4. Interface (implemented in class)
5. Body of class

Objects: It is a basic unit of OOP and represent real life entities. A typical java program creates so many objects.

An object consist of –

1. State.
2. Behaviour.
3. Identity.



Methods :- A method is a collection of statement that perform some specific task and return result to its caller.

Return type

Method name.

```
public   int   max (int x, int y) {
Modifier
    if( x > y) {
        return x;
    } else {
        return y;
    }
}
```

Parameter of method.

Body of method.

Constructor in Java: Constructor are used to initialised the object state. Constructor are also collection of statements (i.e. instruction) that are executed at the time of object creation.

Need of Constructor: Constructor are used to assign the value to the class variables at the time of object creation, either explicitly done by the programmer..

Either constructor is called: Each time an object is created using new() keyword at least one constructor is invoked.

Type of Constructor:

1. Default constructor (No argument)
2. Parameterised Constructor.
3. Copy Constructor.

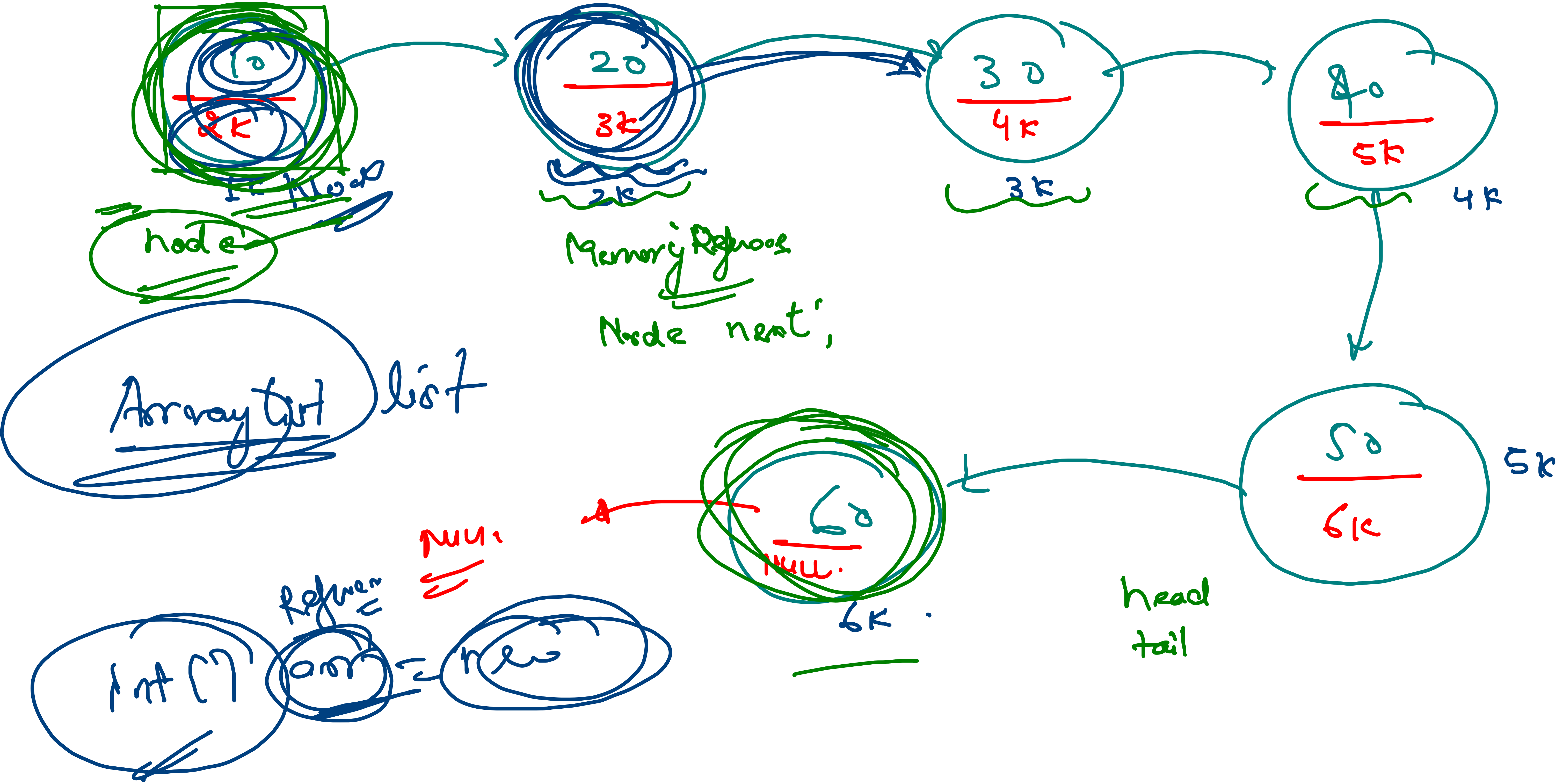
Rules for writing constructor:

* constructor have same name as its class has.
* constructor in java can't be abstract, final, static and synchronized.
* Access modifier can be used in constructor declaration to control its access i.e. which other class can call the constructor.

'this' Reference in Java:

'this' is reference variable that refers to the current object

→ There are so many use of this, but we are not looking for this refer now.


10
2k
20
3k
30
4k
40
5k
50
6k
60
null
1k
2k
3k
4k
5k
6k
node
Memory Reference
Node next;
ArrayList list
null
int[] arr = new
head
tail

LinkedList ⟶

1. list.addFirst(int data);
2. list.addLast(int data);
3. list.addAt(int dat, int pos);

1. list.removeFirst();
2. list.removeLast();
3. list.removeAt(int pos)

Heap array in C++

int* arr = new int[10];

Reference

int arr[10];

Stack

vector<int> arr = new vector();

Add first (10); size = 0

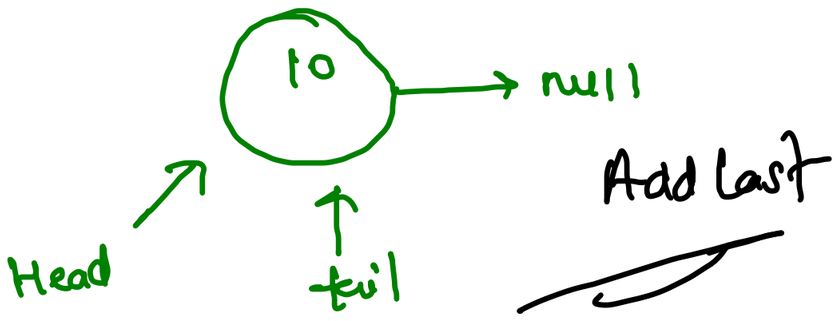
head = null

tail = null

Steps

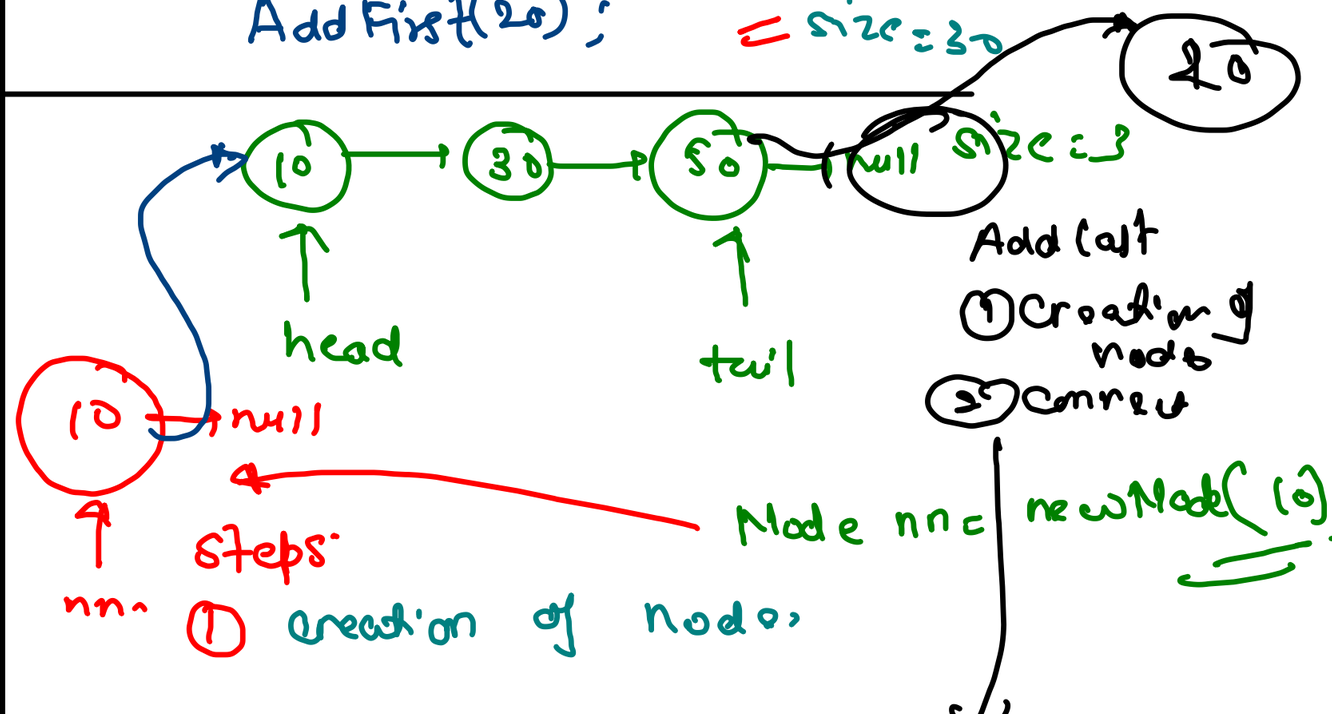
① creation of new node.

② Set head and tail



Add Last

Add First(20); size = 3



Add Last

① Creation of node

② Connect

Node nn = newNode(10);

Steps:

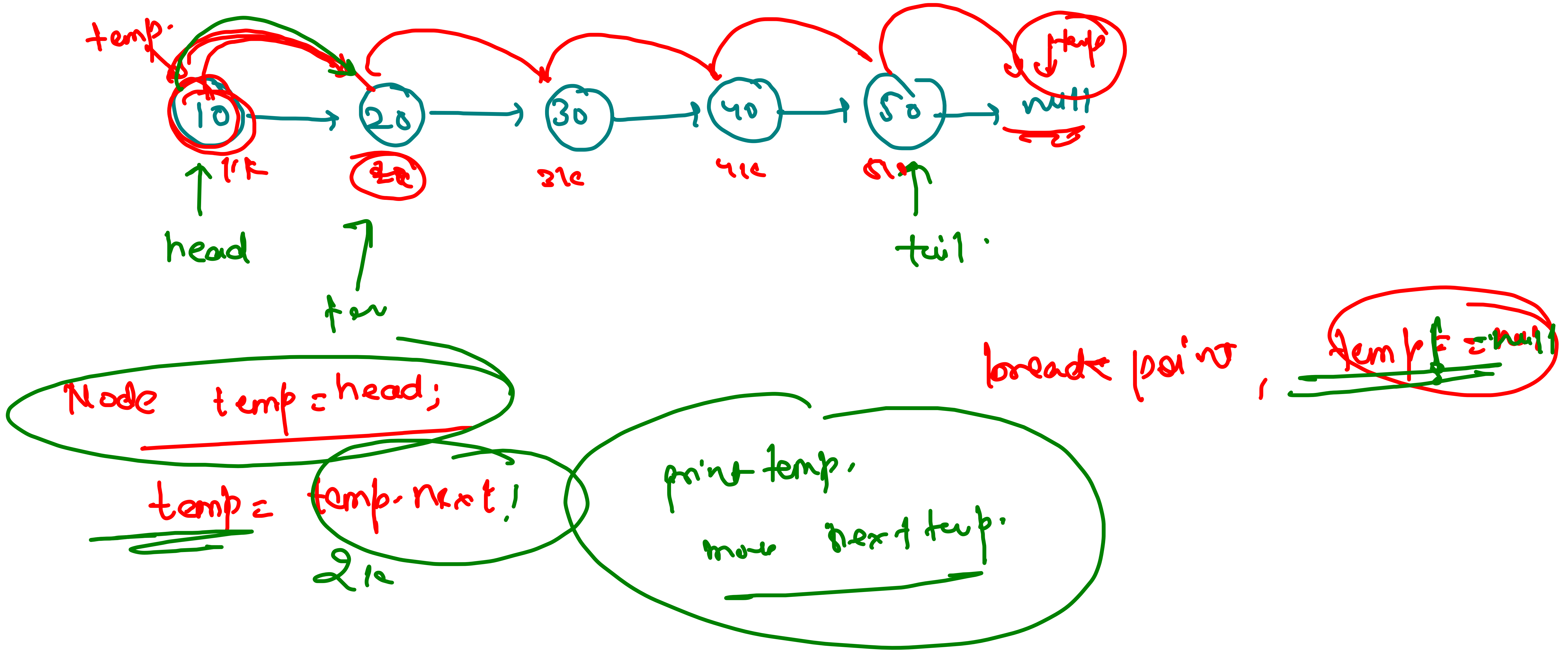
① creation of node.

② Make connection.

{ nn.next = head
  head = nn;

{ ① tail.next = nn
  ② tail = nn;

this.size ++.



temp
10
20
30
40
50
null
head
tail
Node temp = head;
temp = temp.next;
print temp
move next temp
break point , temp = null