Let us start understanding the Linked List data structure.

Linked List is a data structure which consists of sequence of nodes. Each node is connected to the next node and last node’s next is pointing to null, mostly.

Why mostly, because there are variants of Linked Lists, such as

* Single Linked List.
* Single Circular Linked List.
* Double Linked List.
* Circular Double Linked List.
* And others.

Let us try to understand every variant.

Let us start understanding the Linked List data structure.

Linked List is a data structure which consists of sequence of nodes. Each node is connected to the next node and last node’s next is pointing to null, mostly.

Why mostly, because there are variants of Linked Lists, such as

* Single Linked List.
* Single Circular Linked List.
* Double Linked List.
* Circular Double Linked List.
* And others.

Let us try to understand every variant.

Single Linked List

                This is simplest kind of Linked List in which one node is connected to another via a next pointer and last pointer points to null that denotes end of linked list.

[https://4.bp.blogspot.com/-nU-mwQR9ghw/VUczxOVc1pI/AAAAAAAALcA/PYRh8YJTl1I/s320/Singly-linked-list.svg.png](http://4.bp.blogspot.com/-nU-mwQR9ghw/VUczxOVc1pI/AAAAAAAALcA/PYRh8YJTl1I/s1600/Singly-linked-list.svg.png)

  Picture taken from <https://en.wikipedia.org/wiki/File:Singly-linked-list.svg>

              The above diagram shows that first node has data 12 as integer and it is connected to another node via next pointer to node having data 99. The last node contains data 37 which is pointing to null. In this way we terminate Single Linked List.

So if we are supposed to build the structure of Linked List in Java then we can built it using this information.

/\*\*

 \* Class name Node

 \* It denotes the structure of the node.

 \*

 \* int data denotes that the data that the list will handle is of type integer.

 \*

 \* Node next denotes that we will point to next node.

 \*

 \* The constructor is used to create new node and setting node's next to null.

 \* \*/

**public** **class** Node {

**public** **int** data;

**public** Node next;

**public** Node(**int** data) {

**this**.data = data;

              next = **null**;

       }

}