Insert Node in Single Linked List

Algo to insert node in Single Linked List:

Check if root is null

If null then root is new node,return.

If root is not null then capture root in temp node

Now traverse to last node by checking if temp’s next i.e temp.next != null

If the temp.next != null condition is false then temp is pointing to last node.

Set last node’s next as new node.

/\*\*

\* Insert node in List.

\*

\* Let us make sure that first node must be the root of the Linked List

\*

\* If root exists then we will insert the new node at the end of the list.

\*

\* **@param** data the data

\* **@return** the node

\*/

**public** void insertNode(**int** data) {

Node node = **new** Node(data);

**if** (root == **null**) {

root = node;

root.next = **null**;

} **else** {

Node temp = root;

**while** (temp.next != **null**) {

temp = temp.next;

}

temp.next = node;

last = node;

}

}

The insertNode(int data) method defined above takes care of inserting the node in Linked List.

Let us try to understand how this method works line by line.

Node node = **new** Node(data); This statement creates a new node by taking the data of integer type in parameterized constructor of Node class.

**if** (root == **null**) { //If the root is null then this condition is executed.

root = node; //We make sure that first node will be the root node.

root.next = **null**;//Next will be null.

}

**else** {//else condition is executed if root is not null

Node temp = root;//we capture root in temp so as not to move root

**while** (temp.next != **null**) {// if the next pointer of node is not null

temp = temp.next;//traverse to the last node

}

temp.next = node;//temp is last node. Setting temp’s next as new node

last = node;//setting node as last node, optional thing to do.

}