```java
package withoutUsingInnerClassObjext;


public class SortedInsertion {

    Node head;

    class Node{
        int data;
        Node next;

        //creating a constructor for assigning value to data and next as null
        Node(int data){
            this.data = data;
            next = null;
        }

    } // closing of class Node

    // funciton of type Node will return a new node

    Node newNode(int data) {
        Node x = new Node(data);
        return x;
    }
    void sortedInsert(Node new_node) {

        Node curr ;

        if(head == null || head.data > new_node.data) {
            new_node.next = head;
            head = new_node;
        }
        else {  // we are trying to find out the appropriate place for insertion of the new_node
            curr = head;
            while(curr.next!=null && curr.next.data < new_node.data) { // traversing till we find the place to insert
                curr = curr.next; // incrementing by single step
            }
            //insert the new node just after the while loop breaks.
            new_node.next = curr.next;  //careful
            curr.next = new_node;
        }
    }


      void printList() {
          Node curr=head;

      while(curr!=null) {
          System.out.print(curr.data + " -> ");
          curr = curr.next;

      }
      System.out.println("NULL"); // for new line only

```

```
56      }
57
58     public static void main(String[] args) {
59
60         SortedInsertion si = new SortedInsertion();
61         Node new_node;   // creating object of the inner class
62
63         //inserting the first value into the list
64         new_node = si.newNode(4);
65         si.sortedInsert(new_node);
66
67         //2nd value
68         new_node = si.newNode(5);
69         si.sortedInsert(new_node);;
70
71         //3rd value
72         new_node = si.newNode(17);
73         si.sortedInsert(new_node);
74
75         //4th value
76         new_node = si.newNode(21);
77         si.sortedInsert(new_node);
78
79         //5th value
80         new_node = si.newNode(9);
81         si.sortedInsert(new_node);
82
83         si.printList();
84
85
86
87     }
88
89 }
90
```