

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 Tutorial: Linked List

Question 1: Trace the following program code and find the outputs?

```
public class ListNode {
    public char value;
    public ListNode next;
    public ListNode (char v)
    {
        this.value=v;
    }
}
```

```
public class LinkedList {
    private ListNode head=null;
    public void print(){
        ListNode i = head;
        while (i !=null)
        {
            System.out.println (i.value);
            i=i.next;
        }
    }
    public void add(char v, int index){
        ListNode newNode = new ListNode (v); // create node. we need to define constructor in ListNode
        //newNode.value=v;
        if (index==0)
        {
            newNode.next=head;
            head = newNode;
        }
        else {
            ListNode i = head;
            for(int count = 0; count <index-1; count++)
            {
                i= i.next;
            }
            newNode.next = i.next;
            i.next = newNode;
        }
    }
    public void remove(int index){
```

```

        if (index==0){
            head = head.next;          }
        else {
            ListNode i = head;
            for (int count = 0; count <index-1; count++)
            {
                i = i.next;
            }

            ListNode j = i.next;
            i.next = j.next;

        }
    }

    public void isEmpty ()
    {
        if (head != null)
        {
            System.out.println ("LinkedList is not empty");
        }
        else    System.out.println ("LinkedList is empty");
    }
}

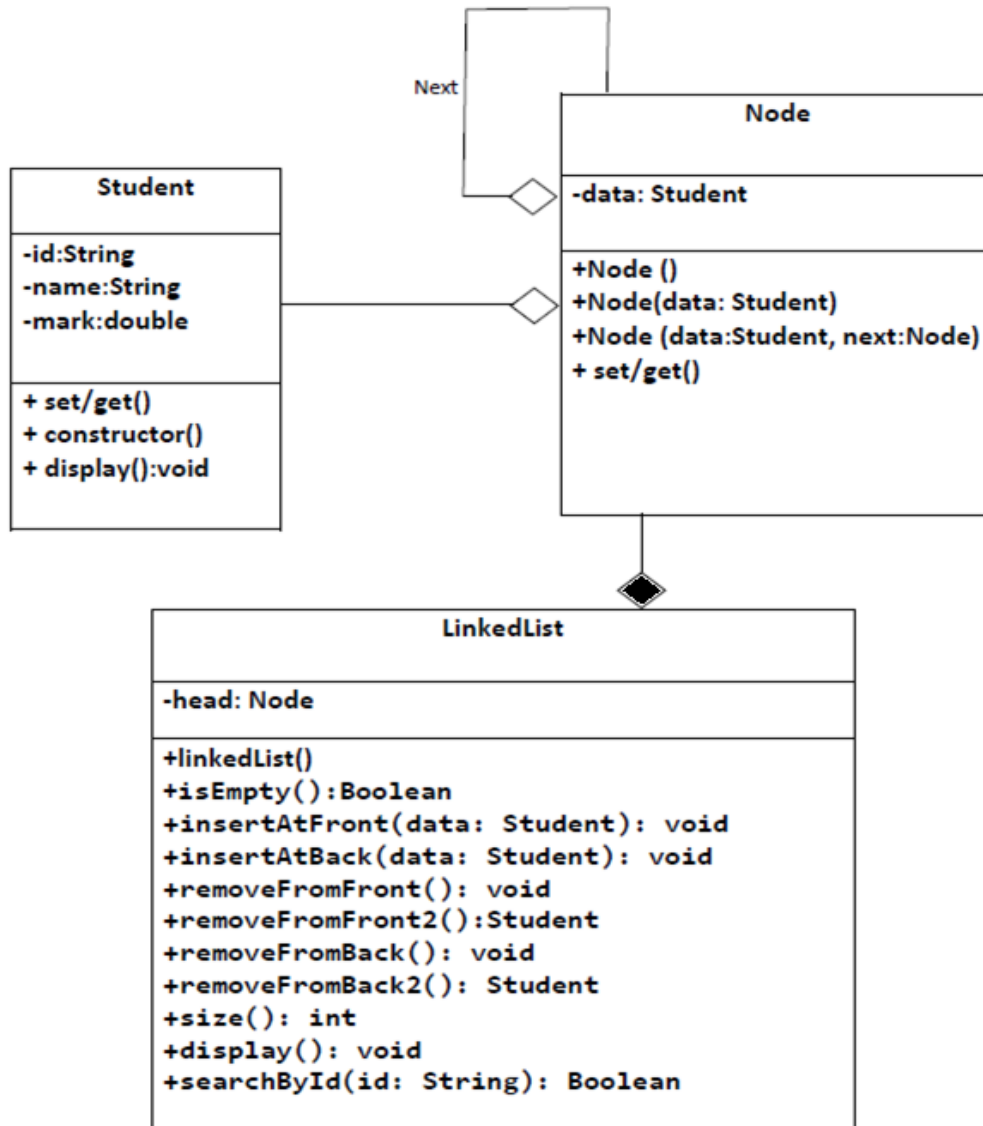
```

```

public class ListTest {
    public static void main(String[] args) {
        LinkedList L = new LinkedList();
        L.add ('a', 0);
        L.add ('b', 1);
        L.add ('c', 2);
        L.add ('h', 1);
        L.add ('x', 0);
        L.add ('z', 5);
        L.remove (1);
        L.remove (3);
        L.remove (3);
        L.remove (0);
        L.remove (1);
        L.remove (0);
        L.isEmpty();
        L.print();
    }
}

```

Question: Consider the following UML and convert into java code:



class LinkedList:

- isEmpty():check if the list is empty and return true otherwise return false

- insertAtFront(data: Student): insert a node of type Student at the front of list
- insertAtBack(data: Student): insert a node of type Student at the end of list
- removeFromFront(): remove the first node from the list
- removeFromFront2(): remove the first node from the list and return its value
- removeFromBack(): remove the last node in the list
- removeFromBack2(): remove the last node in the list and return its value
- size(): this method return the number of nodes in the list
- display(): print all nodes in the list
- searchById(id: String): this method search about the student by id and return true if exist otherwise return false