**Appendix:** **Java codes for Node and NodeStack for reference**

public class Node {  
 private Object element;  
 private Node next;  
 public Node() {  
 this(null, null);  
 }  
   
 public Node(Object e, Node n) {  
 element = e;  
 next = n;  
 }  
 public Object getElement() {  
 return element;   
 }  
 public Node getNext() {   
 return next;  
 }  
 // Modifier methods:  
 public void setElement(Object newElem) {   
 element = newElem;   
 }  
 public void setNext(Node newNext) {  
 next = newNext;   
 }  
}

public class NodeStack {

protected Node head, temp;

protected int number=0;

public NodeStack() {

head = null;

}

public int size() {

return (number);

}

public boolean isEmpty() {

return (number< 1);

}

public void push(Object obj) {

temp=new Node(obj, head);

head=temp;

number=number+1;

}

public Node top() {

if (isEmpty())

System.out.println("Stack is empty.");

return head;

}

public Node pop() {

Node elem;

if (isEmpty()){

System.out.println("Stack is Empty.");

return null;

}

elem = head;

head=head.getNext(); // dereference S[top] for garbage collection.

number=number-1;

return elem;

}

public static void main (String[] args) {

NodeStack S=new NodeStack();

S.push('A');

System.out.println("THe top is: "+(S.top()).getElement());

S.push('B');

System.out.println("THe top is: "+(S.top()).getElement());

}

}