```java
//Saket M Kharche
// Stack implementation using a linked list
class StackLinkedList {
    // Node class represents each element in the linked list
    private class Node {
        int data;  // Data to be stored in the node
        Node next;  // Reference to the next node in the list

        // Constructor to initialize the node
        public Node(int data) {
            this.data = data;
            this.next = null;  // Initially, the next node is null
        }
    }

    private Node top;  // The top node in the stack

    // Constructor to initialize the stack
    public StackLinkedList() {
        top = null;  // Initially, the stack is empty
    }

    // Check if the stack is empty
    public boolean isEmpty() {
        return top == null;  // Stack is empty if top is null
    }

    // Push an element onto the stack
    public void push(int value) {
        Node newNode = new Node(value);  // Create a new node with the given value
        newNode.next = top;  // Point the new node's next to the current top node
        top = newNode;  // Update the top to be the new node
        System.out.println("Pushed " + value + " to the stack.");
    }

    // Pop an element from the stack
    public int pop() {
        if (isEmpty()) {  // If the stack is empty, return -1 indicating failure
            System.out.println("Stack is empty! Cannot pop.");
            return -1;
        } else {
            int value = top.data;  // Get the data of the top node
            top = top.next;  // Move the top pointer to the next node
            return value;  // Return the popped value
        }
    }

    // Peek at the top element of the stack without removing it
    public int peek() {
        if (isEmpty()) {  // If the stack is empty, return -1
            System.out.println("Stack is empty! Cannot peek.");
            return -1;
        } else {
            return top.data;  // Return the data of the top node
        }
    }
}

public class MainLinkedList {
    public static void main(String[] args) {
        StackLinkedList stack = new StackLinkedList();  // Create a new stack using a linked list

        stack.push(10);
        stack.push(20);
        stack.push(30);
        stack.push(40);
        stack.push(50);

        System.out.println("Top element is: " + stack.peek());  // Peek at the top element

        System.out.println("Popped element: " + stack.pop());  // Pop the top element
        System.out.println("Popped element: " + stack.pop());  // Pop another element
    }
}

```