**National University of Computer and Emerging Sciences**



Lab Manual

for

Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Samman Ashraf |
| Lab Instructor(s) | Muhammad Ahmed  Hamza Ayub |
| Section | 4N |
| Semester | Spring 2023 |

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

**Objectives:**

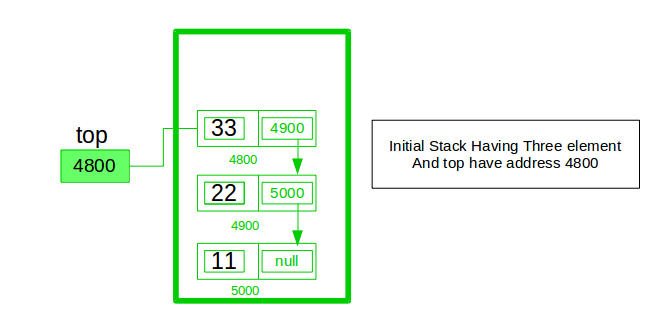
In this lab, students will practice:

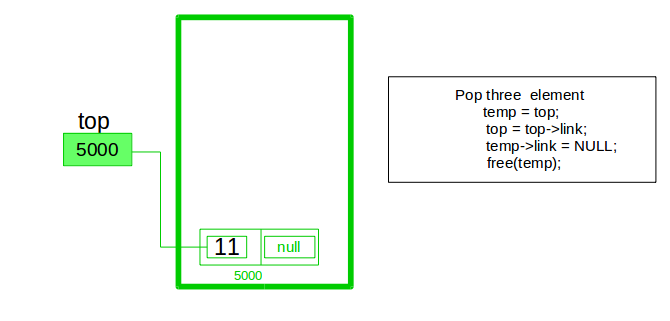
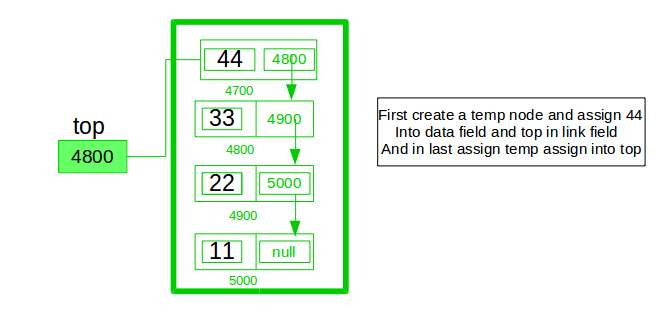
1. Stack with single link-list
2. Circular Link-list

**Introduction**

**Implement a stack using singly linked list**

To implement a stack using singly linked list concept , all the singly linked list operations are performed based on Stack operations LIFO(last in first out) and with the help of that knowledge we are going to implement a stack using single linked list. Using singly linked lists , we implement stack by storing the information in the form of nodes and we need to follow the stack rules and implement using singly linked list nodes . So we need to follow a simple rule in the implementation of a stack which is last in first out and all the operations can be performed with the help of a top variable .Let us learn how to perform Pop , Push , Peek ,Display operations in the following article .





A stack can be easily implemented using the linked list. In stack Implementation, a stack contains a top pointer. which is “head” of the stack where pushing and popping items happens at the head of the list. First node have null in link field and second node link have first node address in link field and so on and last node address in “top” pointer.  
The main advantage of using linked list over an arrays is that it is possible to implement a stack that can shrink or grow as much as needed. In using array will put a restriction to the maximum capacity of the array which can lead to stack overflow. Here each new node will be dynamically allocate. so overflow is not possible.  
**Stack Operations:**

1. **[push()](https://www.geeksforgeeks.org/stack-push-and-pop-in-c-stl/) :** Insert a new element into stack i.e just inserting a new element at the beginning of the linked list.
2. **[pop()](https://www.geeksforgeeks.org/stack-push-and-pop-in-c-stl/) :** Return top element of the Stack i.e simply deleting the first element from the linked list.
3. **[peek()](https://www.geeksforgeeks.org/stack-peek-method-in-java/):** Return the top element.
4. **display():** Print all elements in Stack.

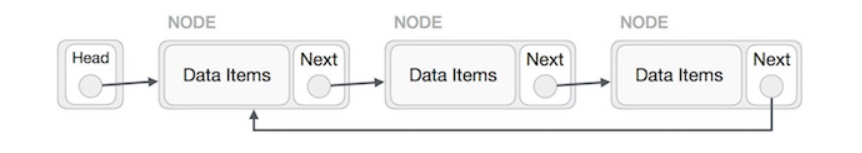
**Circular linked list**

**Introduction:**

Circular Linked List is a variation of Linked list in which the first element points to the last element and the last element points to the first element. Both Singly Linked List and Doubly Linked List can be made into a circular linked list.

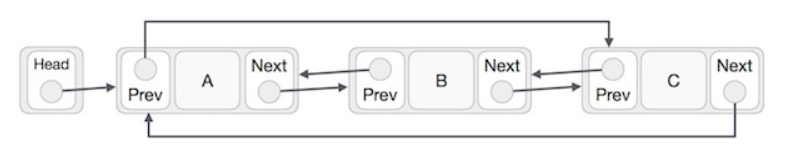
**Singly Linked List as Circular**

In singly linked list, the next pointer of the last node points to the first node.



**Doubly Linked List as Circular**

In doubly linked list, the next pointer of the last node points to the first node and the previous pointer of the first node points to the last node making the circular in both directions.



**Question 1**

1. Implement a template-based stack using a singly linked list. The required member methods are:

**int size()**: returns the count of total element stored in the stack.

**bool isEmpty()**: returns true if the stack is empty else false.

**bool top(T&)**: returns, but does not delete, the topmost element from the stack via the parameter passed by reference. It returns false via a return statement if there is no element in the stack, else it returns true and assigns the top most element to the parameter passed by reference.

**void pop()**: deletes the top most element from the stack. If there is no element, return some error.

**push(T const& e)**: pushes the element “e” on top of the stack.

**Question 2**

# Remove all triple adjacent characters from a string using Stack

**Question 3**

**Delete middle element of a stack**

Given a stack with push(), pop(), empty() operations, delete the middle of it without using any additional data structure.

**Input :** Stack[] = [1, 2, 3, 4, 5]

**Output :** Stack[] = [1, 2, 4, 5]

**Input :** Stack[] = [1, 2, 3, 4, 5, 6]

**Output :** Stack[] = [1, 2, 4, 5, 6]

**Question 4**

**Reverse a circular linked list.**

Given a circular linked list of size n. The problem is to reverse the given circular linked list.

# question