

Lab Report 2 Title: Stack Implementation

Objective: To Implement Stack from Array and Linked List

Functions To Use:

- push(element): Adds an element into the stack
- pop(): Removes an element from the stack
- isEmpty(): Checks if the stack is empty
- isFull(): Checks if the stack is full
- top(): Gives the element at the top

Something About Queue:

A queue is a linear data structure that follows the First-In-First-Out(FIFO) principle, which means that the element that is added first to the queue will be the first one to be removed from it.

In simpler terms, a queue is like a line or queue of people waiting for a service. The first person who joins the line is the first person to be served, and as new people join the line, they add themselves to the back of the back of the line.

In programming, a queue is usually implemented using an array or a linked list. The two main operations on a queue are enqueue and dequeue. Enqueue adds an element to the end of the queue, while dequeue removes the element from the front of the queue. Other operations that can be performed on a queue include peek (to view the front element without removing it), size (to determine the number of elements in the queue), and isEmpty (to check whether the queue is empty or not).

- Main.cpp Code

```
#include "src/ArrayStack.cpp"
#include "src/StackLinkedList.cpp"
#include <iostream>
using namespace std;

int main()
{
    // Taking the size of the stack from the user
    int sizeofarray;
    cout << "Enter The size of stack" << endl;
    cin >> sizeofarray;

    // Creating the stack from array
    ArrayStack stack(sizeofarray);
    int value;
    for (int i = 0; i < sizeofarray; i++)
    {
        cout << "Enter The element to put inside stack" << endl;
        cin >> value;
```

```

        cout << "Adding " << value << " to the stack..." << endl;
        stack.push(value);
    }

    cout << "Adding 5 to the stack...But It Should Print Size limit Reached
since the size of array is " << sizeofarray << endl;
    cout << "Try Adding 5 to the stack..." << endl;
    stack.push(5);

    cout << "Displaying the topmost element of the Stack:" << endl;
    cout << stack.peek() << endl;

    cout << "Removing the topmost element of the Stack one by one:" << endl;
    for (int i = 0; i < sizeofarray; i++)
        cout << stack.pop() << endl;

    cout << "Since all the elemets are removed it cannot return any more" <<
endl;
    cout << stack.pop();

    // Stack from Linked List
    StackLinkedList stackLinked;
    cout << "" << endl;
    cout << "Implementing Stack from Linked List" << endl;
    cout << "" << endl;

    for (int i = 0; i < sizeofarray; i++)
    {
        cout << "Enter The element to put inside stack" << endl;
        cin >> value;
        stackLinked.push(value);
    }

    for (int i = 0; i < sizeofarray; i++)
    {
        cout << "Removing the topmost element of the Stack one by one:" <<
endl;
        cout << "peeking top most element: " << stackLinked.top() << endl;
        cout << "popped element: " << stackLinked.pop() << endl;
    }
}

```

Output

- Array Stack Implementation

```
Enter The size of stack
4
Enter The element to put inside stack
1
Adding 1 to the stack...
Enter The element to put inside stack
2
Adding 2 to the stack...
Enter The element to put inside stack
3
Adding 3 to the stack...
Enter The element to put inside stack
4
Adding 4 to the stack...
Adding 5 to the stack...But It Should Print Size limit Reached since the size of array is
Try Adding 5 to the stack...
Array Size Full
Displaying the topmost element of the Stack:
4
Removing the topmost element of the Stack one by one:
4
3
2
1
Since all the elemets are removed it cannot return any more
Stack is Empty
0
```

- Linked Stack Implementation

```
Implementing Stack from Linked List

Enter The element to put inside stack
4
Enter The element to put inside stack
5
Enter The element to put inside stack
6
Removing the topmost element of the Stack one by one:
peeking top most element: 6
popped element: 6
Removing the topmost element of the Stack one by one:
peeking top most element: 5
popped element: 5
Removing the topmost element of the Stack one by one:
peeking top most element: 4
popped element: 4
```