**Experiment 4.2**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 06/09/2022

**Subject Name:** DAA Lab **Subject Code:** 21-CSP-312

# Aim/Overview of the practical:

Code to push & pop and check Isempty, Isfull and Return top element in stacks using templates.

# Task to be done/ Which logistics used:

Using C++ templates perform push and pop operation on stacks.

# Algorithm/Flowchart:

* Start.
* First we will define the size.
* Then we will create a class template called Stack.
* Then we will check the top of stack using - template <class T> Stack<T>::Stack() { top = -1;
* Then we will push elements into the stack using templates.
* Using template, we will check whether the stack is empty or is full.
* The we will pop an element of stack using templates.
* We will check the top element using template <class T> T Stack<T>::topElement().
* Print the result.
* Stop.

# Steps for experiment/practical/Code:

#include <iostream>

#include <string>

using namespace std;

#define SIZE 5

template <class T> class Stack {

public: Stack();

void push(T k);

T pop();

T topElement();

bool isFull();

bool isEmpty();

private:

int top;

T st[SIZE];

};

template <class T> Stack<T>::Stack() { top = -1; } template <class T> void Stack<T>::push(T k)

{

if (isFull()) {

cout << "Stack is full\n";

}

cout << "Inserted element " << k << endl; top = top + 1;

st[top] = k;

}

template <class T> bool Stack<T>::isEmpty()

{

if (top == -1)

return 1;

else

return 0;

}

template <class T> bool Stack<T>::isFull()

{

if (top == (SIZE - 1)) return 1;

else

return 0;

}

template <class T> T Stack<T>::pop()

{

T popped\_element = st[top];

top--;

return popped\_element;

}

template <class T> T Stack<T>::topElement()

{

T top\_element = st[top]; return top\_element;

} int main()

{

Stack<int> integer\_stack;

Stack<string> string\_stack;

integer\_stack.push(15);

integer\_stack.push(55);

integer\_stack.push(90);

string\_stack.push("Sahil");

string\_stack.push("Kaundal");

cout << integer\_stack.pop() << " is removed from stack" << endl;

cout << string\_stack.pop() << " is removed from stack " << endl;

cout << "Top element is " << integer\_stack.topElement()<< endl;

cout << "Top element is " << string\_stack.topElement()<< endl;

return 0;

}

# Observations/Discussions/ Complexity Analysis:

***Time Complexity:*** O(1)

# Result/Output/Writing Summary:

# 

**Learning outcomes (What I have learnt):**

* Learnt how to implement Stack.
* Learnt how to push an element in Stack.
* Learnt how to pop an element in Stack.
* Learnt how to check an element in Stack.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |