|  |  |  |  |
| --- | --- | --- | --- |
| **Branch: Instrumentation & Control Engineering** | | **Year:** Second Year | |
| **Division: C** | **Roll No: 04** | **GR Number: 11911180** | **Subject:** OOPS |
| **Assignment No: 6** | **Date of Submission: 29-04-2021** | **Student Full Name: Shaunak Sudhir Deshpande** | |

Aim: Implement all types of Stack using concept of template.

Software Used: MinGW, VSCode

Code:

#include<iostream>

#include<string>

using namespace std;

template <class T>

class Stack

{

    public:

    Stack();

    void push(T i);

    T pop();

    private:

    int top;

    T st[100];

};

template <class T>

Stack<T>::Stack()

{

    top=-1;

}

template <class T>

void Stack<T>::push(T i)

{

 st[++top]=i;

}

template <class T>

T Stack<T>::pop()

{

    return st[top--];

}

int main()

{

    Stack<int> int\_stack;

    Stack<string> str\_stack;

    int\_stack.push(444);

    str\_stack.push("OOPS Subject");

    str\_stack.push("Welcome to");

    cout<<int\_stack.pop()<<endl;

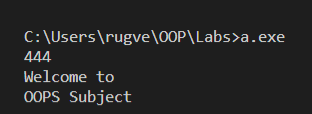
    cout<<str\_stack.pop()<<endl;

    cout<<str\_stack.pop()<<endl;

    return 0;

}

Output:



Analysis of Program:

In this program, we are trying to demonstrate the use of template classes in C++ with the use of Stacks.

Stacks can be of multiple types, integer, float, long, string, character etc. In this case , we are creating a template class for stacks, and then creating stacks of different types such as int, str, etc, without having to write separate classes of these types.

Conclusion:

Templates are a very useful feature in C++ / Object oriented programming and helps us to reduce the amount of work that we need to do in order to write code which can be of multiple types.