

Lab 11

1. Write a C++ program to implement the Stack (PUSH and POP) operations using Template. Consider the inputs of different data types (e.g.: integer, float, string, etc.) to verify the functionalities.

A. Code:-

```
#include <iostream>
#define SIZE 2
#include <string>
using namespace std;
template <class T> class Stack
{
private:
    int top;
    T stack[];
public:
    Stack()
    {
        top = -1;
    }
    void push(T ele)
    {
        if (isFull())
        {
            cout << "STACK OVERFLOW\n";
        }
        else
        {
            stack[++top] = ele;
            cout<<ele<<" was pushed in the stack."<<endl;
        }
    }
    void pop()
    {
        if (isEmpty())
```

```

    {
        cout << "STACK UNDERFLOW\n";
    }
    else
    {
        cout<<stack[top--]<<" was popped."<<endl;
    }
}
bool isFull()
{
    if (top == (SIZE - 1))
        return 1;
    else
        return 0;
}
bool isEmpty()
{
    if (top == -1)
        return 1;
    else
        return 0;
}
};
int main()
{
    cout<<"The size of the stacks is taken as 2\n"<<endl;
    Stack<int> a;
    Stack<float> b;
    Stack<char> c;
    Stack<string> d;
    a.push(2);
    a.push(7);
    a.push(9);
    a.pop();
    a.pop();
    a.pop();
    printf("\n");
    b.push(2.56);
    b.push(7.42);
    b.push(9.89);

```

```
b.pop();
b.pop();
b.pop();
printf("\n");
c.push('H');
c.push('E');
c.push('Y');
c.pop();
c.pop();
c.pop();
return 0;
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:

The size of the stacks is taken as 2.

2 was pushed in the stack.

7 was pushed in the stack.

STACK OVERFLOW

7 was popped.

2 was popped.

STACK UNDERFLOW

2.56 was pushed in the stack.

7.42 was pushed in the stack.

STACK OVERFLOW

7.42 was popped.

2.56 was popped.

STACK UNDERFLOW

H was pushed in the stack.

E was pushed in the stack.

STACK OVERFLOW

E was popped.

H was popped.

STACK UNDERFLOW

2. Write an exception handling programming in C++ to
a) Demonstrate divide-by-zero operation (e.g., $A/(B-C)$)

A. Code:-

```
#include<iostream>
using namespace std;
int main()
{
    int a, b, c;
    float res;
    cout << "Enter the value of a: ";
    cin>>a;
    cout << "Enter the value of b: ";
    cin>>b;
    cout << "Enter the value of c: ";
    cin>>c;
    try
    {
        if ((b-c)!=0)
        {
            res = (float) a/(b-c);
            cout << "Result is: " << res;
        }
        else
        {
            throw (b - c);
        }
    }
    catch (int i)
    {
        cout << "Answer is infinite because a-b is:" << i<< endl;
        cout << "MathError: Divide by zero";
    }
    return 0;
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:

```
Enter the value of a: 6
Enter the value of b: 7
Enter the value of c: 6
Result is: 6
```

b) Array out of index operation

A. Code:-

```
#include<iostream>
using namespace std;
int main()
{
    int n, ele, loc;
    cout<<"Enter the size of the array: ";
    cin>>n;
    int arr[n];
    cout<<"Enter the location where you want to insert element: ";
    cin>>loc;
    try
    {
        if (loc<5 && loc>=0)
        {
            cout << "Enter the element: ";
            cin>>ele;
            arr[loc] = ele;
            cout << "Element Inserted..."<<endl;
        }
        else
        {
            throw 0;
        }
    }
    catch (int i)
    {
        cout <<"Error: Array out of Index."<< endl;
    }
    return 0;
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:

```
Enter the size of the array: 3
Enter the location where you want to insert element: 2
Enter the element: 56
Element Inserted...
```

3. Write a C++ program to throw multiple exceptions and define multiple catch statements.

A. Code:-

```
#include<iostream>
using namespace std;
int main()
{
    int x;
    cout << "Enter a number: ";
    cin>>x;
    try
    {
        if (x < 0)
            throw x;
        else
            throw 'x';
    }
    catch (int x)
    {
        cout << "The number is less than zero";
    }
    catch (char x)
    {
        cout << "The number is greater than or equal to zero";
    }
    return 0;
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:

```
Enter a number: 53
The number is greater than or equal to zero
```