CU
CHANDIGARH UNIVERSITY

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Design and Analysis of Algorithms

Experiment 1.1

23CSP-301

Submitted To:
Faculty Name: Er. Moh. Shaqlain

Submitted By:
Name: Garvi Dabas
UID: 23BCS11346
Section: KRG - 2B
Semester: 5th

Aim:

To analyze if the stack is empty or full, and if elements are present, return the top element in the stack using templates. Also, perform push and pop operations on the stack.

Algorithm:

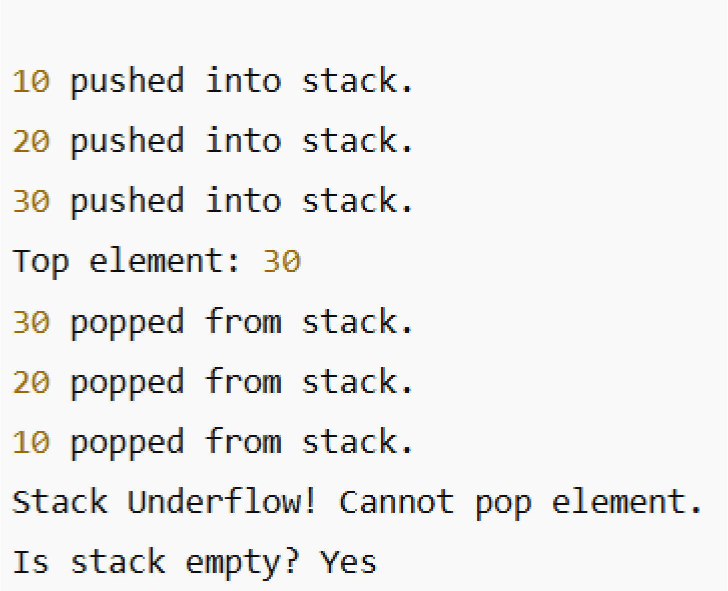
1. Start the program.
2. Create a class Stack using templates.
3. Initialize stack with maximum size and top = -1.
4. Implement the following functions:
5. isEmpty() → returns true if stack is empty.
6. isFull() → returns true if stack is full.
7. push() → inserts an element if stack is not full.
8. pop() → removes an element if stack is not empty.
9. peek() → returns the top element without removing it.
10. Perform push, pop, and peek operations.
11. Display results.
12. Stop the program.

Code:

Answer 1:

```
#include <iostream>
using namespace std;

template <class T>
class Stack {
    int top;
    int size;
    T *arr;

public:
    // Constructor
    Stack(int s) {
        size = s;
        arr = new T[size];
        top = -1;
    }

    // Destructor
    ~Stack() {
        delete[] arr;
    }

    // Check if stack is empty
    bool isEmpty() {
        return (top == -1);}
    // Check if stack is full
    bool isFull() {
        return (top == size - 1);
    }

    // Push operation
    void push(T element) {
        if (isFull()) {
            cout << "Stack Overflow! Cannot push " << element << endl;
        } else {
            arr[++top] = element;
            cout << element << " pushed into stack." << endl;
        }
    }

    // Pop operation
    void pop() {
        if (isEmpty()) {
            cout << "Stack Underflow! Cannot pop element." << endl;
        } else {
            cout << arr[top--] << " popped from stack." << endl;
        }
    }

    // Peek operation
    T peek() {
        if (isEmpty()) {
            cout << "Stack is empty! No top element." << endl;
            return -1; // return dummy value
        } else {
            return arr[top];
        }
    }
};

// Driver code
int main() {
    Stack<int> s(5); // stack of integers

    s.push(10);
    s.push(20);
    s.push(30);

    cout << "Top element: " << s.peek() << endl;

    s.pop();
    s.pop();
    s.pop();
    s.pop(); // extra pop to check underflow

    cout << "Is stack empty? " << (s.isEmpty() ? "Yes" : "No") << endl;
    return 0;
}
```

Output 1:



```
10 pushed into stack.
20 pushed into stack.
30 pushed into stack.
Top element: 30
30 popped from stack.
20 popped from stack.
10 popped from stack.
Stack Underflow! Cannot pop element.
Is stack empty? Yes
```

Conclusion:

The stack was successfully implemented using C++ templates. The program checks for underflow and overflow, performs push and pop operations, and returns the top element.