Riphah International University

Data Structure (Lab)

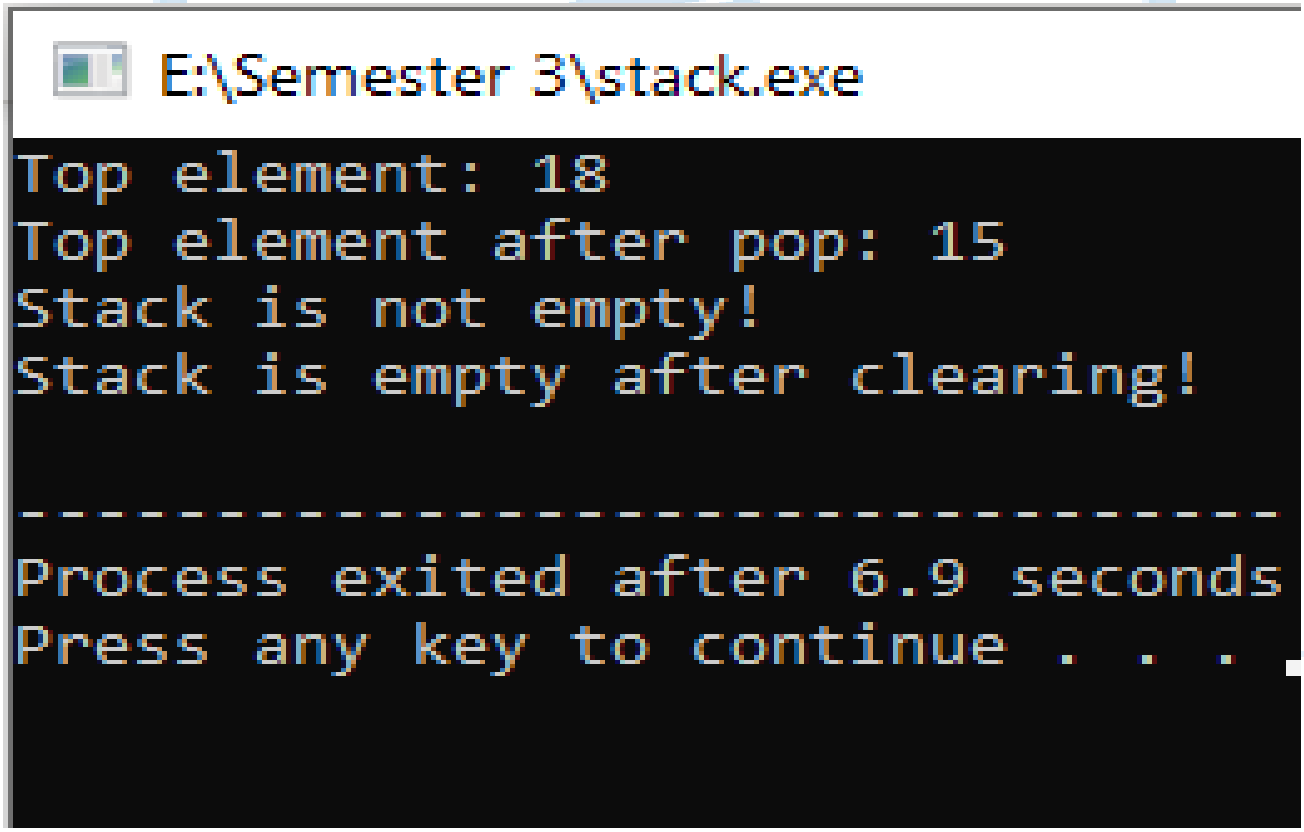
Lab Task 4th

Name: Mir Ahmad Shah
Sap id: 54906
Instructor: Mr. Zeeshan Ali
Section: BSCS (3-1)
Submission Date: September 20, 2024

1:

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

class Stack {
private:
    int* stackArray;
    int capacity;
    int top;

public:
    Stack(int ignored = 0) {
        capacity = 100;
        stackArray = new int[capacity];
        top = -1;
    }

    void push(int dataItem) {
        if (top >= capacity - 1) {
            cout << "Stack overflow, cannot add more elements!" << endl;
            return;
        }
        stackArray[++top] = dataItem;
    }

    void pop() {
        if (isEmpty()) {
            cout << "Stack underflow, no elements to pop!" << endl;
            return;
        }
        top--;
    }

    int peek() {
        if (isEmpty()) {
            cout << "Stack is empty, nothing to peek!" << endl;
            return -1;
        }
        return stackArray[top];
    }

    void clear() {
        top = -1;
    }

    bool isEmpty() {
        return (top == -1);
    }
};

int main() {
```

```
    Stack myStack;

    myStack.push(5);
    myStack.push(15);
    myStack.push(18);

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

    myStack.pop();

    cout << "Top element after pop: " << myStack.peek() << endl;

    if (myStack.isEmpty()) {
        cout << "Stack is empty!" << endl;
    } else {
        cout << "Stack is not empty!" << endl;
    }

    myStack.clear();

    if (myStack.isEmpty()) {
        cout << "Stack is empty after clearing!" << endl;
    }

    return 0;
}
```



2:

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

class Stack {
private:
    char* stackArray;
    int capacity;
    int top;

public:
    Stack(int size) {
        capacity = size;
        stackArray = new char[capacity];
        top = -1;
    }
    void push(char dataItem) {
        if (top >= capacity - 1) {
            cout << "Stack overflow, cannot add more elements!" << endl;
            return;
        }
        stackArray[++top] = dataItem;
    }

    char pop() {
        if (isEmpty()) {
            cout << "Stack underflow, no elements to pop!" << endl;
            return '\0';
        }
        return stackArray[top--];
    }

    bool isEmpty() {
        return (top == -1);
    }
};

string reverseStringUsingStack(string str) {
    Stack s(str.length());

    for (char ch : str) {
        s.push(ch);
    }

    string reversedStr = "";
```

```
    while (!s.isEmpty()) {
        reversedStr += s.pop();
    }

    return reversedStr;
}

int main() {
    string str;
    cout << "Enter a string to reverse: ";
    getline(cin, str);

    string reversedStr = reverseStringUsingStack(str);
    cout << "Reversed string: " << reversedStr << endl;

    return 0;
}
```