Week 14-15

Q3. Program with class template with single parameter

#include <iostream>

using namespace std;

template <typename T>

class Stack {

private:

T\* stackArray;

int top;

int capacity;

public:

Stack(int size = 10) {

stackArray = new T[size];

top = -1;

capacity = size;

}

~Stack() {

delete[] stackArray;

}

void push(T element) {

if (top == capacity - 1) {

cout << "Error: Stack Overflow" << endl;

return;

}

stackArray[++top] = element;

}

T pop() {

if (top == -1) {

cout << "Error: Stack Underflow" << endl;

return T();

}

return stackArray[top--];

}

bool isEmpty() const {

return top == -1;

}

};

int main() {

Stack<int> intStack;

intStack.push(10);

intStack.push(20);

cout << "Popped element: " << intStack.pop() << endl;

Stack<double> doubleStack;

doubleStack.push(3.14);

doubleStack.push(2.71);

cout << "Popped element: " << doubleStack.pop() << endl;

return 0;

}