Rajvaibhav Rahane

17u283 223045

SE-C Comp,Viit,Pune

***CODE:***

#include<iostream>

#include<exception>

using namespace std;

class EmptyStackException:public exception{

public:

const char\*what()const throw(){

return "EmptyStackException";

}

};

class FullStackException:public exception{

public:

const char\*what()const throw(){

return "FullStackException";

}

};

template<class T>

class Stack{

private:

T \*stk;

int stk\_size;

int tos;

public:

Stack(int stk\_size=10){

if(stk\_size<0)

throw std::bad\_array\_new\_length();

stk=new T[stk\_size];

this->stk\_size=stk\_size;

tos=-1;

}

void push(T element);

T pop();

void printStack();

bool isFull(){

return tos==stk\_size-1;

}

bool isEmpty(){

return tos==-1;

}

~Stack(){

delete stk;

cout<<"destructor called\n";

}

};

template<class T>

void Stack<T>::push(T element){

if(isFull()){

throw FullStackException();

}

else{

\*(stk+(++tos))=element;

}

}

template<class T>

T Stack<T>::pop(){

if(isEmpty()){

throw EmptyStackException();

}

return \*(stk+(tos--));

}

template<class T>

void Stack<T>::printStack(){

int tempTos=tos;

while(tempTos>-1){

cout<<\*(stk+(tempTos--))<<" ";

}

cout<<endl;

}

int main(){

Stack<int> myStack(9);

for(int i=1;i<=11;i++){

cout<<"Try to add Add:"<<i<<endl;

try{

myStack.push(i);

myStack.printStack();

}catch(exception& e){

cout<<e.what()<<" was thrown\n";

}

}

myStack.printStack();

for(int i=1;i<=11;i++){

try{

cout<<"Pop:"<<myStack.pop()<<endl;

}catch(exception& e){

cout<<e.what()<<" was thrown"<<endl;

}

}

return 0;

}

***Output:***

