Source Code

/\*

Name: Eranus Thompson

Using a queue to simulate a stack

\*/

#include <iostream>

#include <queue>

#include <string>

using namespace std;

template<class T>

class Stack

{

public:

Stack() { // create an empty stack

}

bool empty() { // returns true if the stack is empty

if (items.empty())

return true;

return false;

}

void push(T item) {// adds item to the top of stack

T temp;

if (items.empty()) {

items.push(item);

}

else {

while (!items.empty()) {

temp = items.front();

aux.push(temp);

items.pop();

}

items.push(item);

}

while (!aux.empty()) {

items.push(aux.front());

aux.pop();

}

}

void pop(){// removes an item from the stack

if (!items.empty()) {

items.pop();

}

}

T top(){// returns at the top of the stack

T temp;

if (!items.empty())

temp = items.front();

return temp;

}

private:

queue<T> items; // holds the items you push (in reverse)

queue<T> aux; // helpful for pop and top

};

int main() {

Stack <int> num;

Stack <string> name;

for (int i = 0; i < 10; i++) { //push items to stack

num.push(i + 1);

}

while (!num.empty()) { //prints out the items in stack

cout << num.top() << " ";

num.pop();

}

cout << endl;

name.push("cat");

name.push("dog");

name.push("frog");

name.push("fish");

while (!name.empty()) {

cout << name.top() << " ";

name.pop();

}

cout << endl;

return 0;

}

/\*

Input : 1 2 3 4 5 6 7 8 9 10

Output: 10 9 8 7 6 5 4 3 2 1

Inputv: cat dog frog fish

Output: fish frog dog cat

\*/

