**Day 06**

/\*assignment:Exception

create Stack class for storing 10 numbers

create function push(int number)--->number will get stored in array

create function pop() will return top most number ,last added number

Note:Handle StackFull and StackEmpty Exception

Stack s;//array[4];

s.push(10);s.push(20);s.push(30);//s.push(40); s.push(50);

int a=s.pop();//------> 30

a=s.pop();//----->20

a=s.pop();//------>10

s.pop();//\*/

#include <iostream>

#include <string>

using namespace std;

class StackException {

public:

string message;

StackException(string message) {

this->message = message;

}

};

class Stack {

private:

int arr[10];

int top;

public:

Stack() {

top = -1;

}

void push(int n) {

if (top >= 9) {

throw StackException("Overflow");

} else {

arr[++top] = n;

}

}

int pop() {

if (top < 0) {

throw StackException("Underflow");

}

return arr[top--];

}

void display() {

cout << "Array : " << endl;

for (int i = top; i >=0 ; i--) {

cout << arr[i] << " ";

}

cout << endl;

}

};

int main() {

Stack s;

try {

s.push(1);

s.push(2);

s.push(3);

s.push(4);

s.push(5);

s.push(6);

s.push(7);

s.push(8);

s.push(9);

s.push(10);

s.display();

s.push(11);

s.display();

int pop = s.pop();

pop = s.pop();

pop = s.pop();

pop = s.pop();

pop = s.pop();

pop = s.pop();

pop = s.pop();

pop = s.pop();

s.display();

} catch (const StackException& e) {

cout << "Exception: " << e.message << endl;

}

return 0;

}

