1. Create a custom exception class named StackException. The Push()and Pop() method should throw object of StackException when the stack is full or empty respectively.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class MyStack

{

private int[] StackArr;

private int top;

private int max;

public MyStack(int size)

{

StackArr = new int[size];

top = -1;

max = size;

}

public void push(int item)

{

if (top == max - 1)

{

throw new StackException("Stack OverFlow!!!");

}

else

{

StackArr[++top] = item;

}

}

public int pop()

{

if (top == -1)

{

throw new StackException("Err: No element to pop");

}

else

{

Console.WriteLine("Popped Element : " + StackArr[top]);

return StackArr[top--];

}

}

public void display()

{

if (top == -1)

{

throw new StackException("Stack is Empty");

}

else

{

Console.WriteLine("Inserted items are: ");

for (int i = 0; i <= top; i++)

{

Console.WriteLine("Item[" + (i + 1) + "]: " + StackArr[i]);

}

}

}

}

class Program1

{

static void Main()

{

MyStack stack = new MyStack(5);

try

{

stack.push(155);

stack.push(228);

stack.push(980);

stack.push(682);

stack.push(64);

stack.push(5);

stack.push(19);

stack.push(35);

}

catch (StackException se)

{

Console.WriteLine(se);

}

try

{

//stack.pop();

}

catch (StackException se)

{

Console.WriteLine(se);

}

try

{

stack.display();

}

catch (StackException se)

{

Console.WriteLine(se);

}

Console.ReadLine();

}

}

public class StackException : Exception

{

public StackException(string s) : base(s)

{

}

}

}

