**ASSIGNMENT 4**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class exceptions

{

private int[] ele;

private int top;

private int max;

public class StackException\_push : Exception

{

public override string Message

{

get

{

return "Stack Overflow, can't perform push operation further";

}

}

}

public class StackException\_pop : Exception

{

public override string Message

{

get

{

return "Stack is Empty, can't perform pop operation further";

}

}

}

public exceptions(int size)

{

ele = new int[size];

top = -1;

max = size;

}

public void push(int item)

{

if (top == max - 1)

{

throw new StackException\_push();

}

else

{

ele[++top] = item;

}

}

public int pop()

{

if (top == -1)

{

throw new StackException\_pop();

}

else

{

Console.WriteLine("Popped element is " + ele[top]);

return ele[top--];

}

}

public void printStack()

{

if (top == -1)

{

Console.WriteLine("Stack is Empty");

return;

}

else

{

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

{

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

}

}

}

}

class Prog

{

public static void Main(string[] args)

{

try

{

exceptions e = new exceptions(5);

e.push(100);

e.push(200);

e.push(300);

e.push(400);

// e.push(500);

//e.push(600);

//e.push(700);

Console.WriteLine("Items are : ");

e.printStack();

e.pop();

e.pop();

e.pop();

e.pop();

e.pop();

// e.pop();

// e.pop();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.ReadKey();

}

}