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.

**INPUT:**

**Popping from empty array**

using System;

using System.Collections;

class StackException : Exception

{

public StackException() { }

public StackException(string exceptionMessage) : base(exceptionMessage)

{

}

}

public class StackAssignment

{

static void push(int num, ref Stack stack)

{

try

{

if (stack.Count < 5)

{

stack.Push(num);

}

else

{

throw new StackException("StackOverflow Exception: Stack is Full");

}

}

catch (StackException e) // We could be catching anything here

{

Console.WriteLine(e.Message);

}

}

static void pop(ref Stack stack)

{

try

{

if (stack.Count == 0)

{

throw new StackException("popException: Stack is Empty ");

}

else

{

stack.Pop();

}

}

catch (StackException e) // We could be catching anything here

{

Console.WriteLine(e.Message);

}

}

public static void Main(string[] args)

{

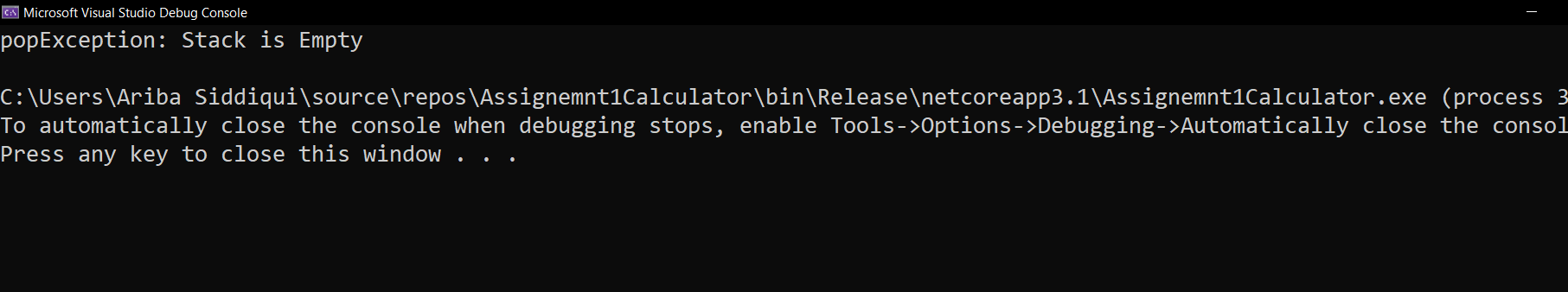
Stack stack = new Stack();

pop(ref stack);

}

}

**OUTPUT:**



**StackOverFlow Exception:**

using System;

using System.Collections;

class StackException : Exception

{

public StackException() { }

public StackException(string exceptionMessage) : base(exceptionMessage)

{

}

}

public class StackAssignment

{

static void push(int num, ref Stack stack)

{

try

{

if (stack.Count < 5)

{

stack.Push(num);

}

else

{

throw new StackException("StackOverflow Exception: Stack is Full");

}

}

catch (StackException e) // We could be catching anything here

{

Console.WriteLine(e.Message);

}

}

static void pop(ref Stack stack)

{

try

{

if (stack.Count == 0)

{

throw new StackException("popException: Stack is Empty ");

}

else

{

stack.Pop();

}

}

catch (StackException e) // We could be catching anything here

{

Console.WriteLine(e.Message);

}

}

public static void Main(string[] args)

{

Stack stack = new Stack();

push(1, ref stack);

push(5, ref stack);

push(3, ref stack);

push(7, ref stack);

push(3, ref stack);

push(77, ref stack);

}

}

**OUTPUT:**

