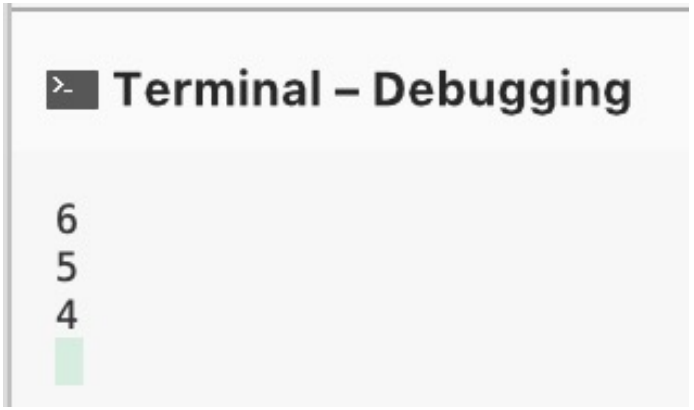
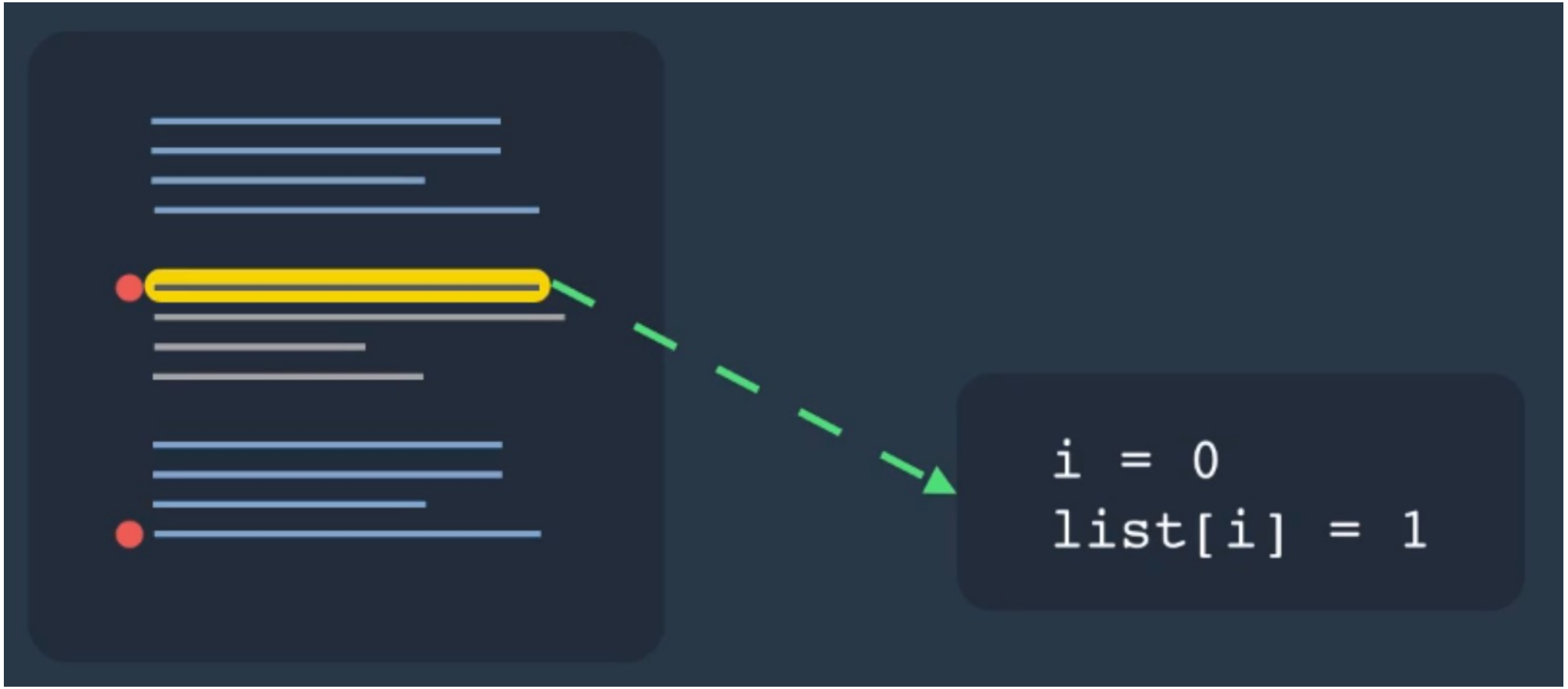
1. Debugging Tools in Visual Studio

```
using System;
using System.Collections.Generic;

namespace Debugging
{
    class Program
    {
        static void Main(string[] args)
        {
            var numbers = new List<int> {1, 2, 3, 4, 5, 6};
            var smallests = GetSmallests(numbers, 3);

            foreach (var number in smallests)
            {
                Console.WriteLine(number);
            }
        }

        public static List<int> GetSmallests(List<int> list, int count)...

        public static int GetSmallest(List<int> list)...
    }
}
```

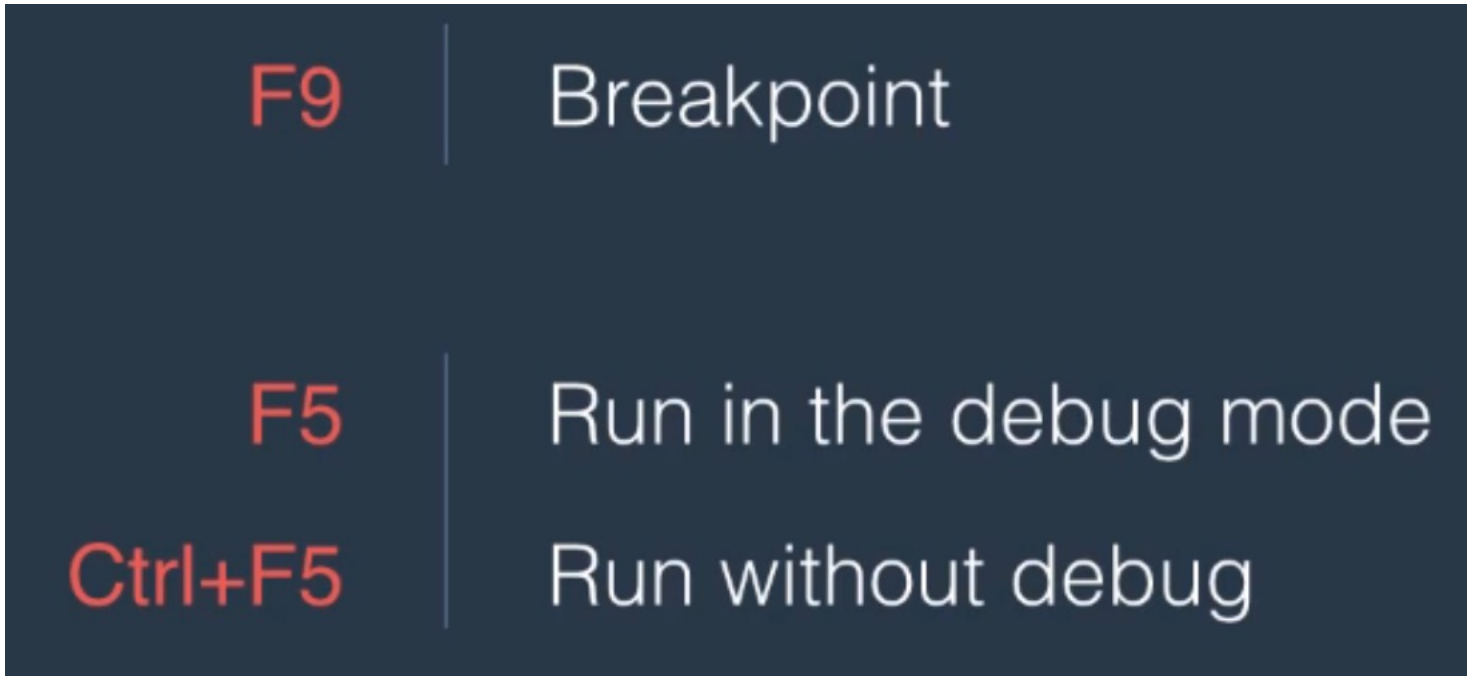
```
public static List<int> GetSmallests(List<int> list, int count)
{
    var smallests = new List<int>();

    while (smallests.Count < count)
    {
        var min = GetSmallest(list);
        smallests.Add(min);
        list.Remove(min);
    }

    return smallests;
}

public static int GetSmallest(List<int> list)
{
    //Assume that first number is minimum.

    var min = list[0];
    for (var i = 1; i < list.Count; i++)
    {
        if (list[i] > min)
        {
            min = list[i];
        }
    }
    return min;

}
```



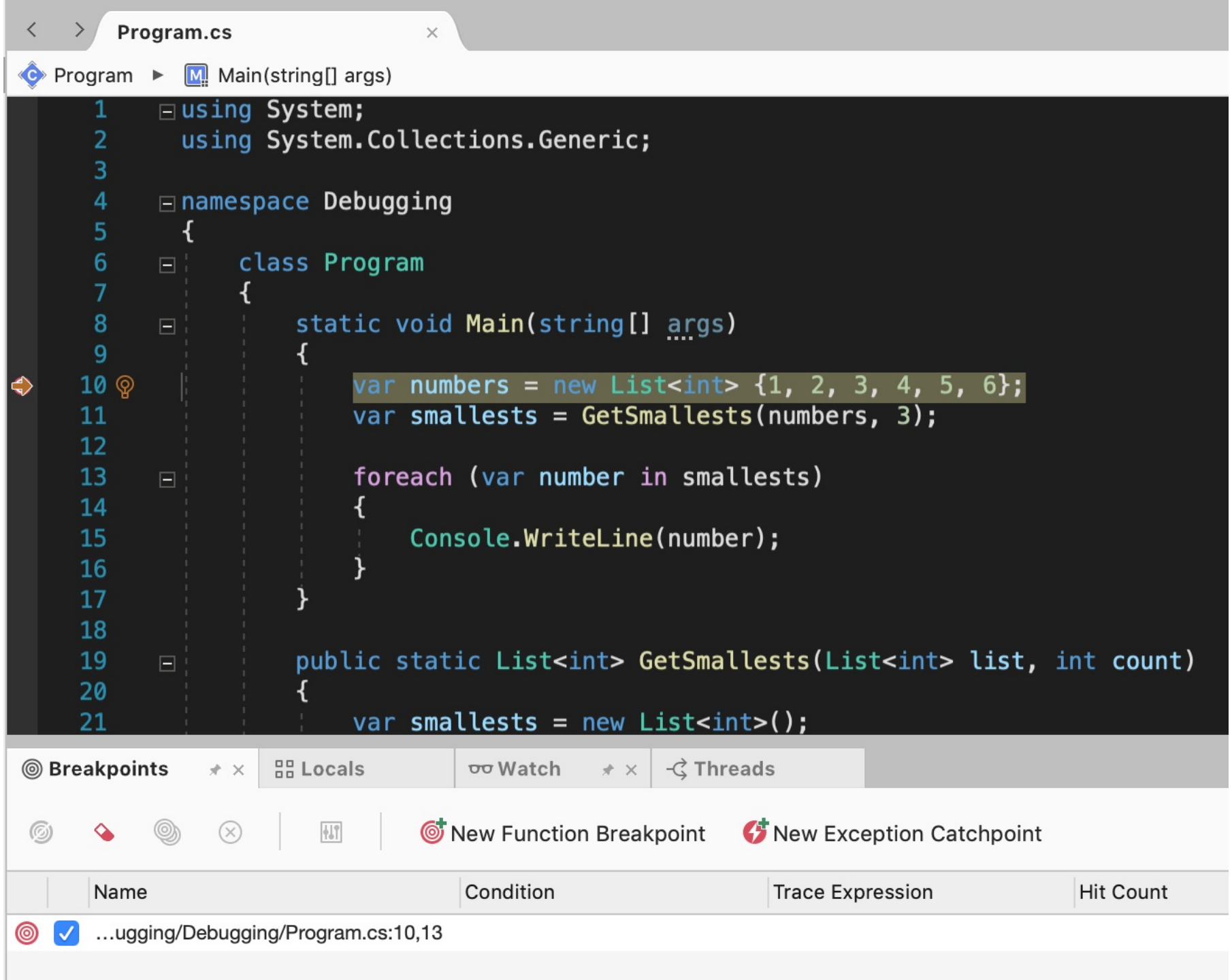


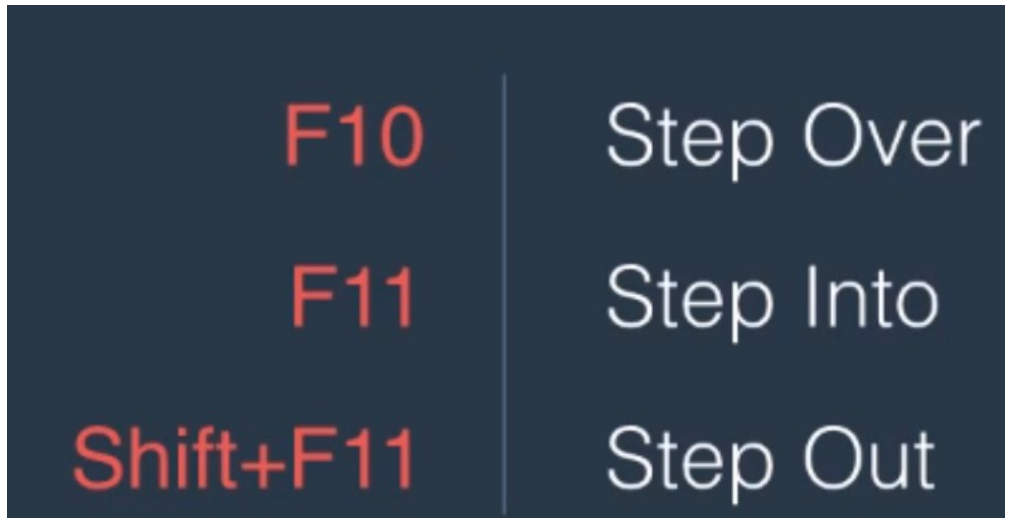
How does debugging work in terms of the process. First you need to put one or more breakpoints in your application and then run your application in the debug mode.

When you run your application into debug mode the execution stops on your breakpoint and there you can inspect the values of different variables to make sure they're holding the right value.
And if not you can change your code.

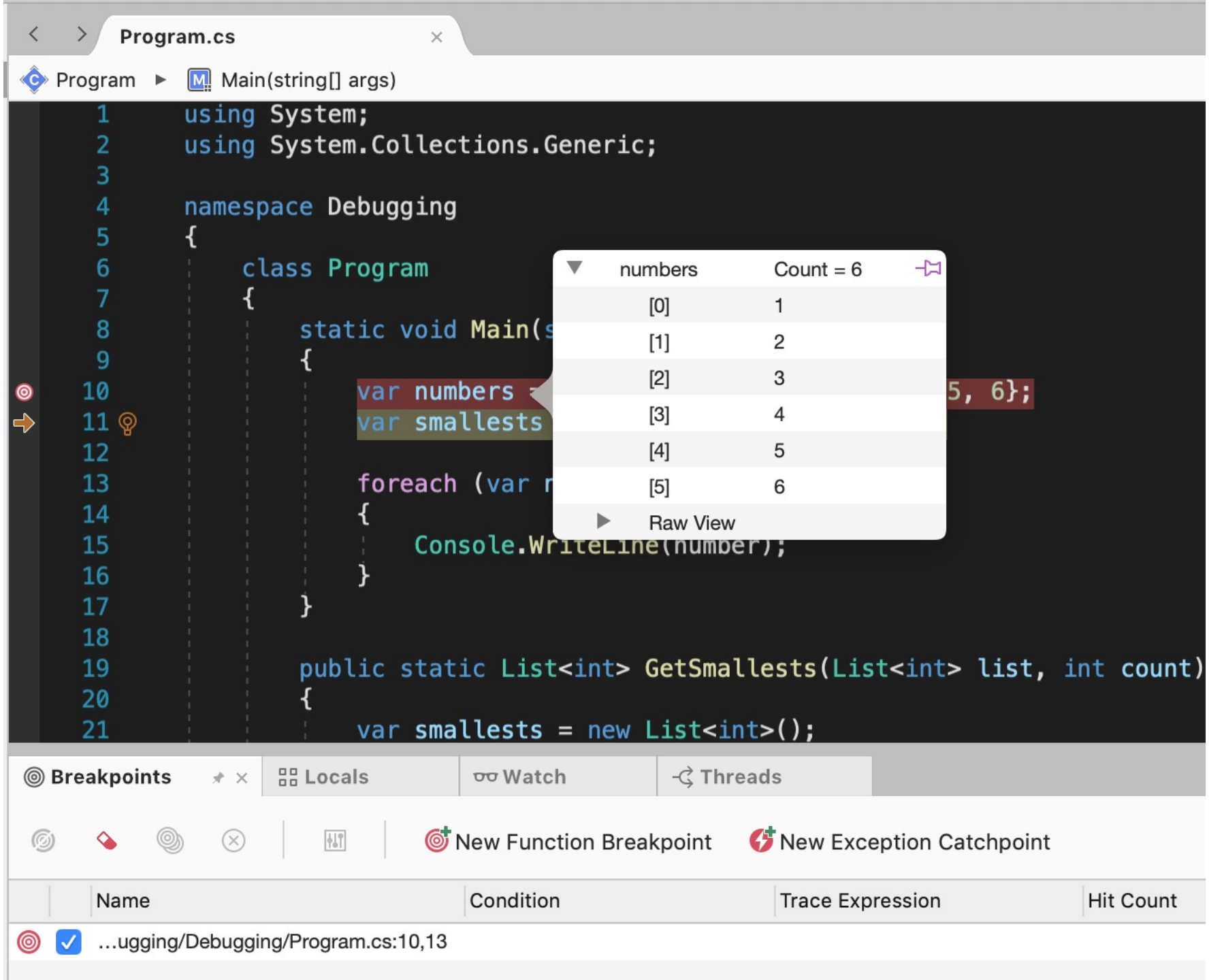

F9
Breakpoint
F5
Run in the debug mode
Ctrl+F5
Run without debug

```
1   using System;
2   using System.Collections.Generic;
3
4   namespace Debugging
5   {
6       class Program
7       {
8           static void Main(string[] args)
9           {
10              var numbers = new List<int> {1, 2, 3, 4, 5, 6};
11              var smallests = GetSmallests(numbers, 3);
12
13              foreach (var number in smallests)
14              {
15                  Console.WriteLine(number);
16              }
17          }
18
19          public static List<int> GetSmallests(List<int> list, int count)
20          {
21              var smallests = new List<int>();
22
23              while (smallests.Count < count)
24              {
25                  var min = GetSmallest(list);
26                  smallests.Add(min);
27                  list.Remove(min);
28              }
29
30              return smallests;
31          }
```


Program.cs
Program
Main(string[] args)
using System;
using System.Collections.Generic;
namespace Debugging
{
class Program
{
static void Main(string[] args)
{
var numbers = new List<int> {1, 2, 3, 4, 5, 6};
var smallests = GetSmallests(numbers, 3);
foreach (var number in smallests)
{
Console.WriteLine(number);
}
}
public static List<int> GetSmallests(List<int> list, int count)
{
var smallests = new List<int>();
Breakpoints
Locals
Watch
Threads
New Function Breakpoint
New Exception Catchpoint
Name
Condition
Trace Expression
Hit Count
...ugging/Debugging/Program.cs:10,13


F10
Step Over
F11
Step Into
Shift+F11
Step Out

F5	Run in debug mode
Shift+F5	Stop the debug mode


Program.cs
Program
Main(string[] args)
using System;
using System.Collections.Generic;
namespace Debugging
{
class Program
{
static void Main(s
{
var numbers =
5, 6};
var smallests
foreach (var n
{
Console.WriteLine(number);
}
}
public static List<int> GetSmallests(List<int> list, int count)
{
var smallests = new List<int>();
numbers
Count = 6
[0] 1
[1] 2
[2] 3
[3] 4
[4] 5
[5] 6
Raw View
Breakpoints
Locals
Watch
Threads
New Function Breakpoint
New Exception Catchpoint
Name
Condition
Trace Expression
Hit Count
…ugging/Debugging/Program.cs:10,13

```
1   using System;
2   using System.Collections.Generic;
3
4   namespace Debugging
5   {
6       class Program
7       {
8           static void Main(string[] args)
9           {
10              var numbers = new List<int> {1, 2, 3, 4, 5, 6};
11              var smallests = GetSmallests(numbers, 3);
12
13              foreach (var number in smallests)
14              {
15                  Console.WriteLine(number);
16              }
17          }
18
19          public static List<int> GetSmallests(List<int> list, int count)
20          {
21              var smallests = new List<int>();
22
23              while (smallests.Count < count)
24              {
25                  var min = GetSmallest(list);
26                  smallests.Add(min);
27                  list.Remove(min);
```

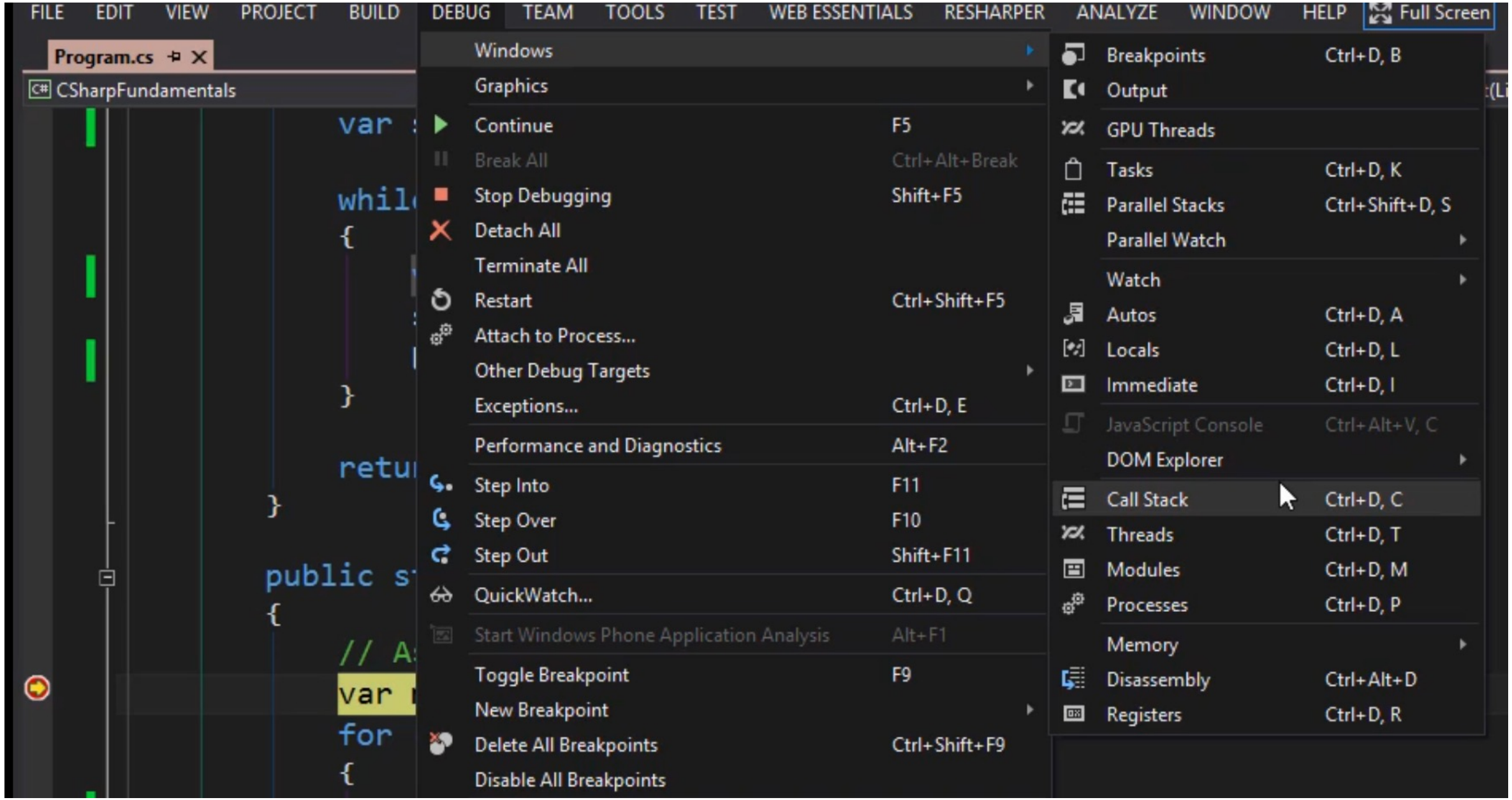
Terminal – Debugging

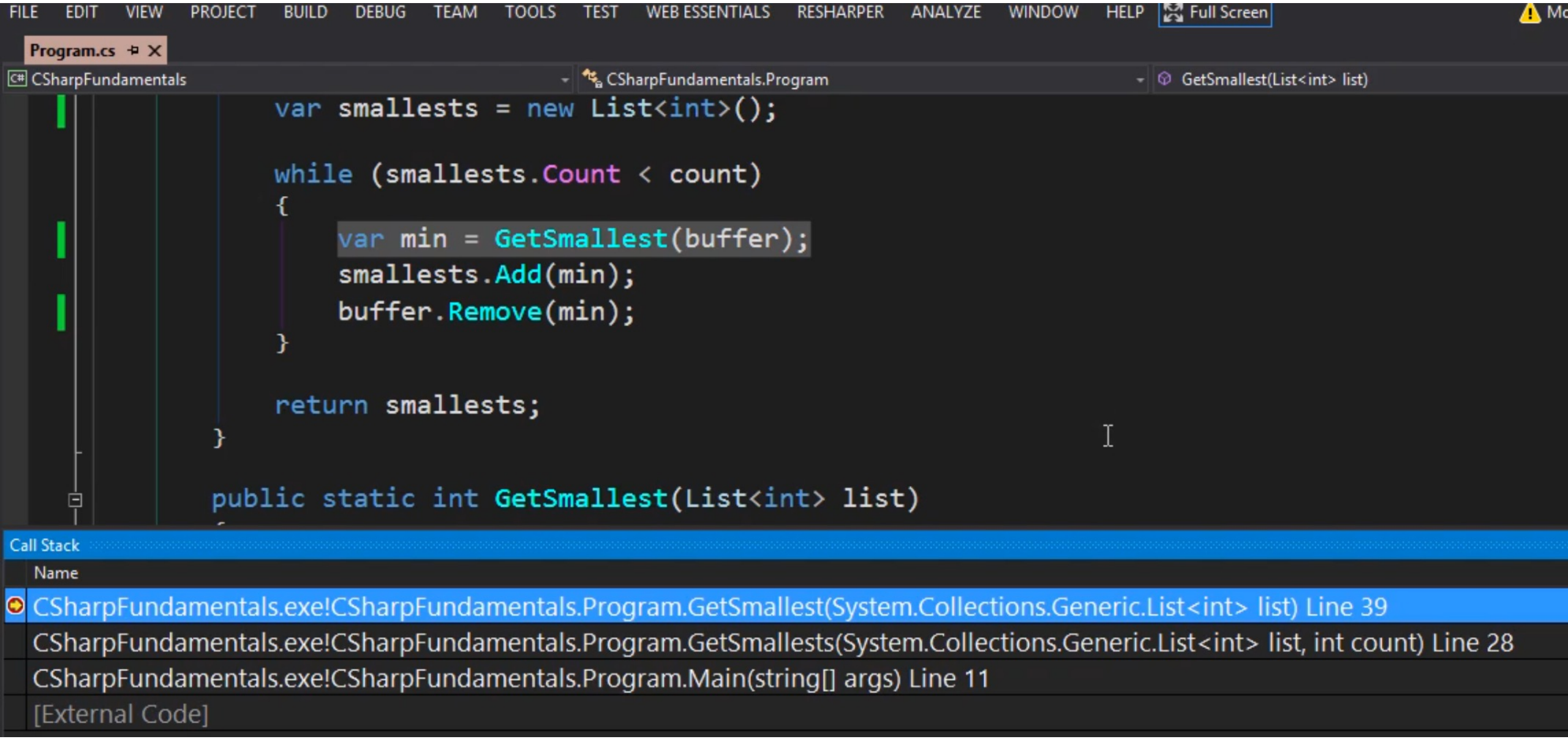
```
1
2
3
```

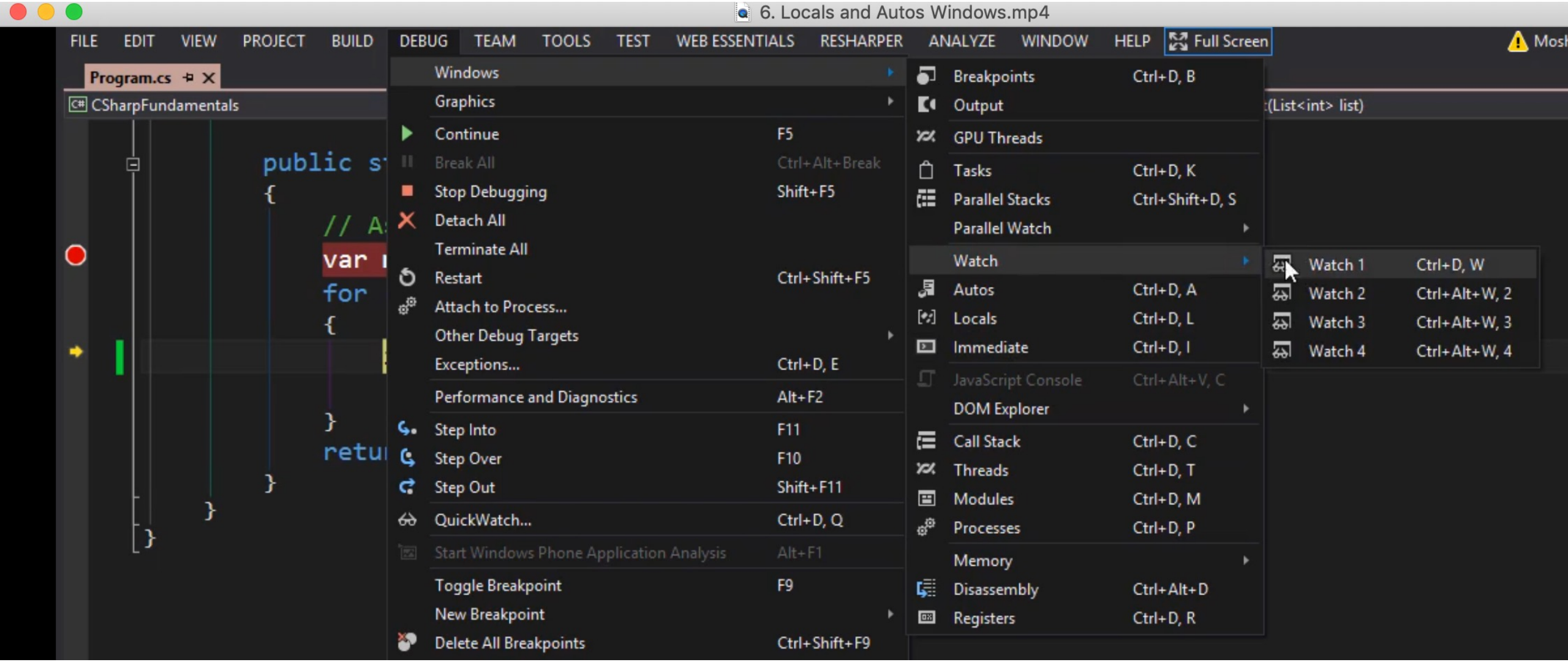
F9	Breakpoint
F5	Run in the debug mode
Ctrl+F5	Run without debug

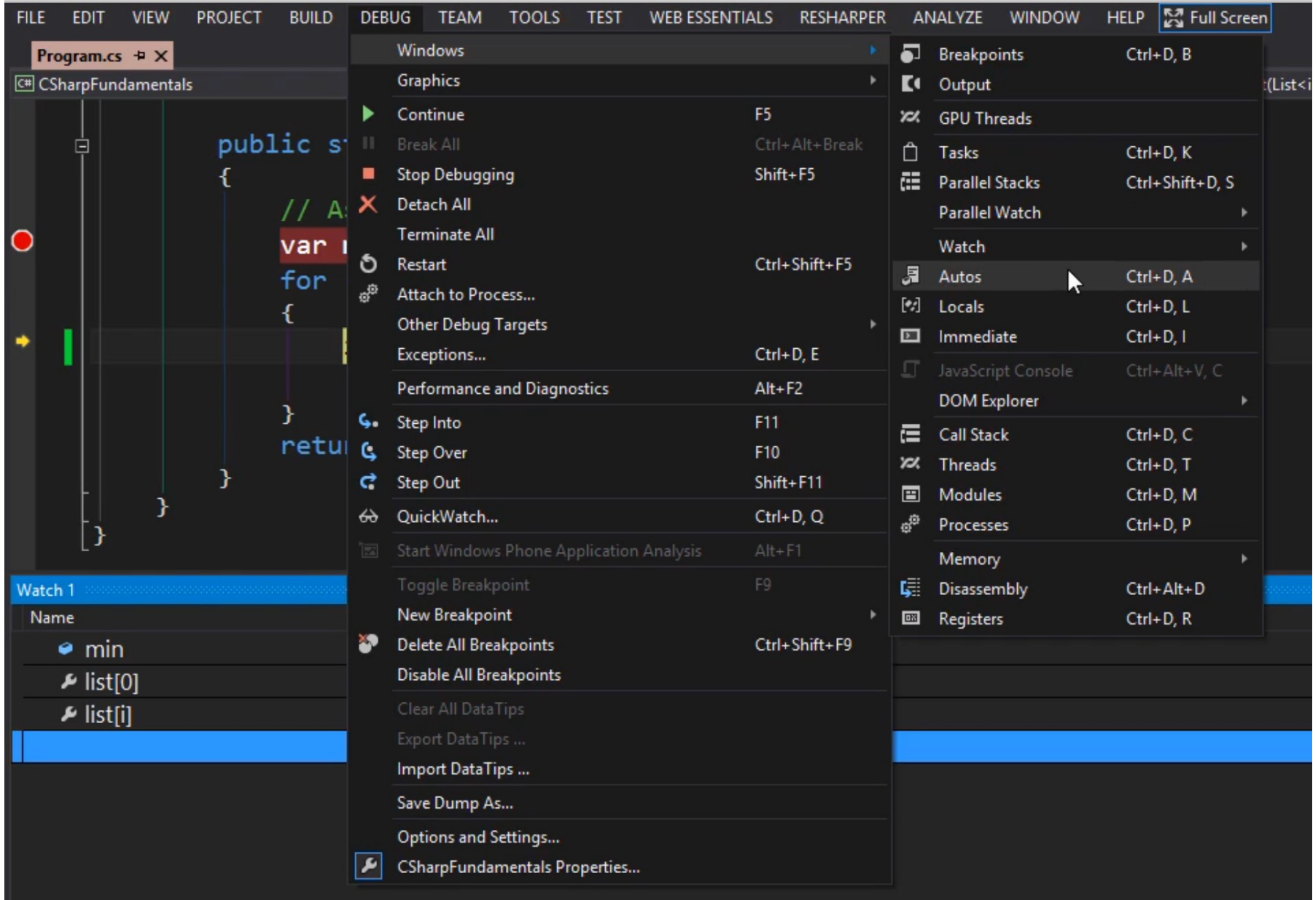
F10	Step Over
F11	Step Into
Shift+F11	Step Out

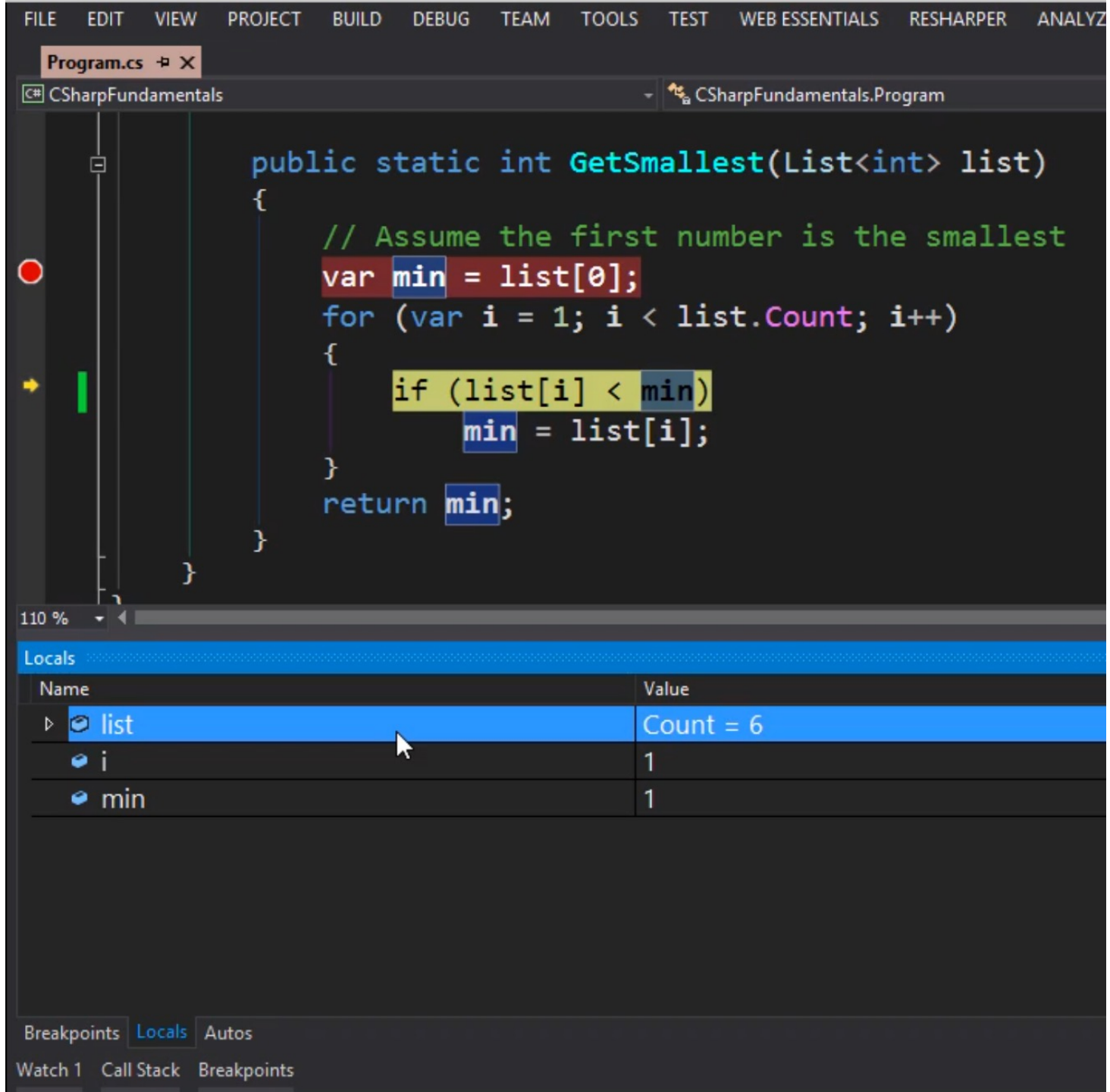
F5	Run in debug mode
Shift+F5	Stop the debug mode


FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZE WINDOW HELP Full Screen
Program.cs
CSharpFundamentals
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+D, E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Ctrl+D, Q
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Disable All Breakpoints
Breakpoints Ctrl+D, B
Output
GPU Threads
Tasks Ctrl+D, K
Parallel Stacks Ctrl+Shift+D, S
Parallel Watch
Watch
Autos Ctrl+D, A
Locals Ctrl+D, L
Immediate Ctrl+D, I
JavaScript Console Ctrl+Alt+V, C
DOM Explorer
Call Stack Ctrl+D, C
Threads Ctrl+D, T
Modules Ctrl+D, M
Processes Ctrl+D, P
Memory
Disassembly Ctrl+Alt+D
Registers Ctrl+D, R


FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZE WINDOW HELP Full Screen
Program.cs
CSharpFundamentals
CSharpFundamentals.Program
GetSmallest(List<int> list)
var smallests = new List<int>();
while (smallests.Count < count)
{
var min = GetSmallest(buffer);
smallests.Add(min);
buffer.Remove(min);
}
return smallests;
}
public static int GetSmallest(List<int> list)
Call Stack
Name
CSharpFundamentals.exe!CSharpFundamentals.Program.GetSmallest(System.Collections.Generic.List<int> list) Line 39
CSharpFundamentals.exe!CSharpFundamentals.Program.GetSmallests(System.Collections.Generic.List<int> list, int count) Line 28
CSharpFundamentals.exe!CSharpFundamentals.Program.Main(string[] args) Line 11
[External Code]

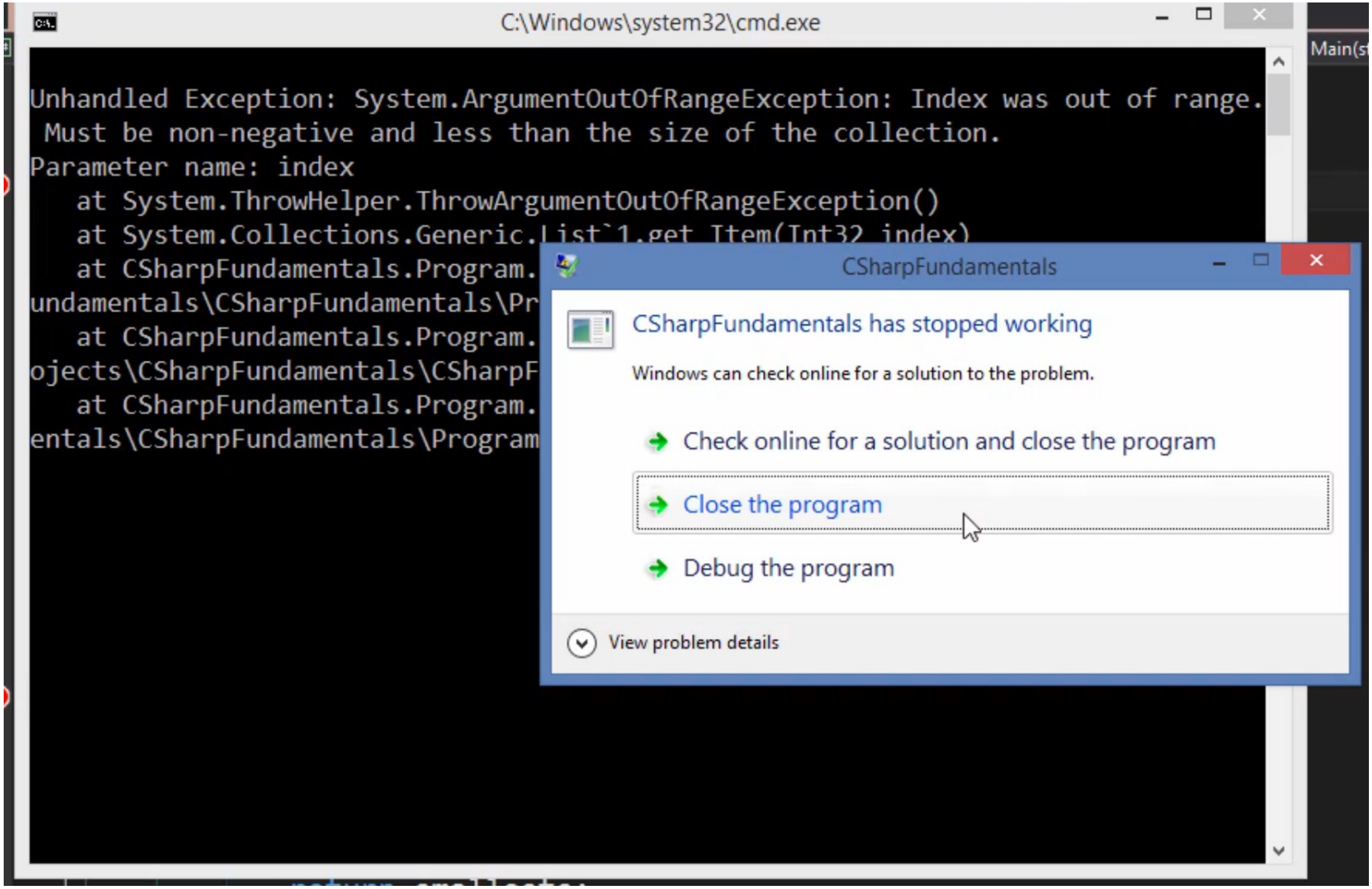

6. Locals and Autos Windows.mp4
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZE WINDOW HELP Full Screen
Program.cs
CSharpFundamentals
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+D, E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Ctrl+D, Q
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Breakpoints Ctrl+D, B
Output
GPU Threads
Tasks Ctrl+D, K
Parallel Stacks Ctrl+Shift+D, S
Parallel Watch
Watch
Autos Ctrl+D, A
Locals Ctrl+D, L
Immediate Ctrl+D, I
JavaScript Console Ctrl+Alt+V, C
DOM Explorer
Call Stack Ctrl+D, C
Threads Ctrl+D, T
Modules Ctrl+D, M
Processes Ctrl+D, P
Memory
Disassembly Ctrl+Alt+D
Registers Ctrl+D, R
Watch 1 Ctrl+D, W
Watch 2 Ctrl+Alt+W, 2
Watch 3 Ctrl+Alt+W, 3
Watch 4 Ctrl+Alt+W, 4

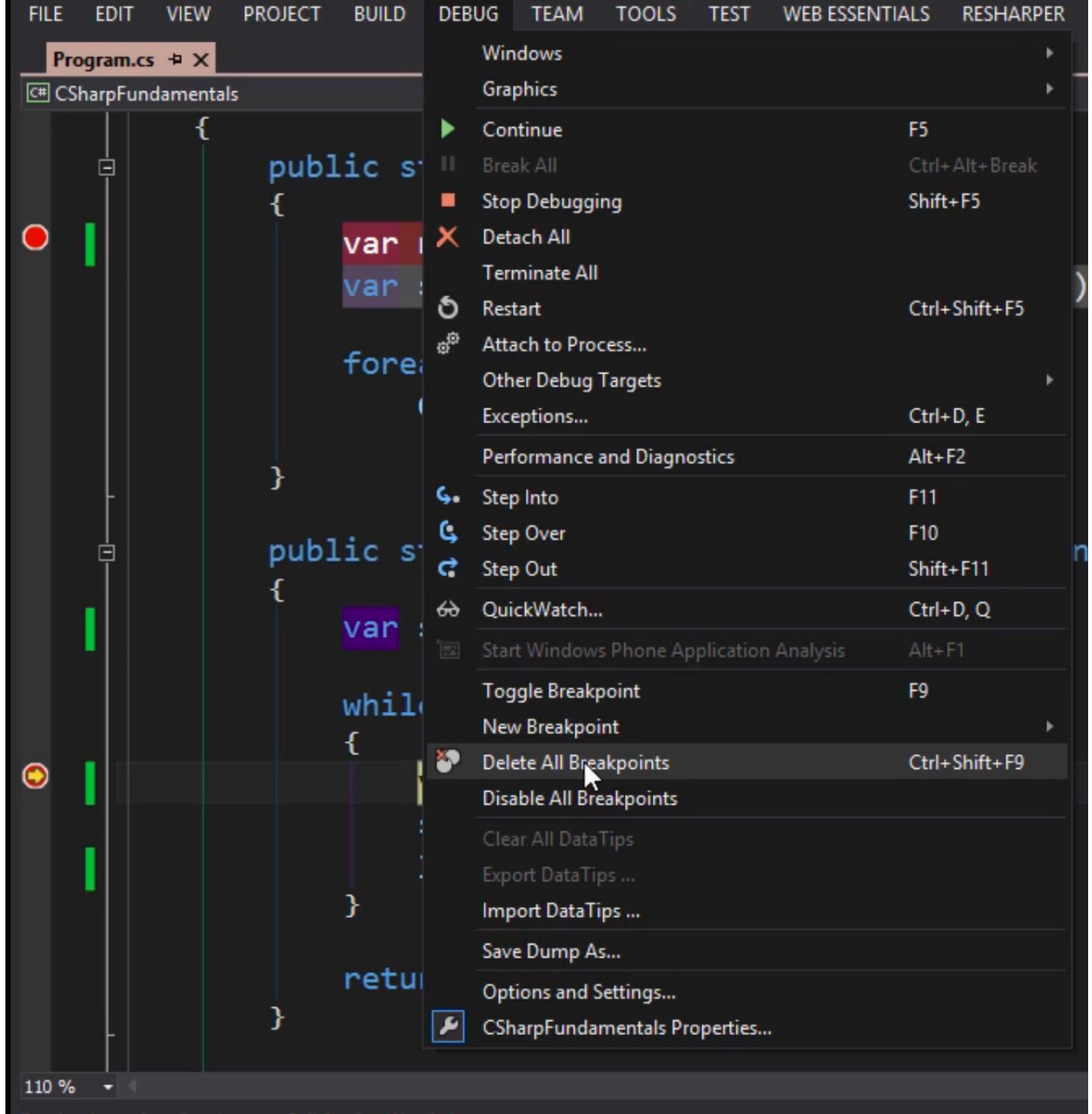

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZE WINDOW HELP Full Screen
Program.cs
CSharpFundamentals
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+D, E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Ctrl+D, Q
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Disable All Breakpoints
Clear All DataTips
Export DataTips ...
Import DataTips ...
Save Dump As...
Options and Settings...
CSharpFundamentals Properties...
Breakpoints Ctrl+D, B
Output
GPU Threads
Tasks Ctrl+D, K
Parallel Stacks Ctrl+Shift+D, S
Parallel Watch
Watch
Autos Ctrl+D, A
Locals Ctrl+D, L
Immediate Ctrl+D, I
JavaScript Console Ctrl+Alt+V, C
DOM Explorer
Call Stack Ctrl+D, C
Threads Ctrl+D, T
Modules Ctrl+D, M
Processes Ctrl+D, P
Memory
Disassembly Ctrl+Alt+D
Registers Ctrl+D, R
Watch 1
Name
min
list[0]
list[i]

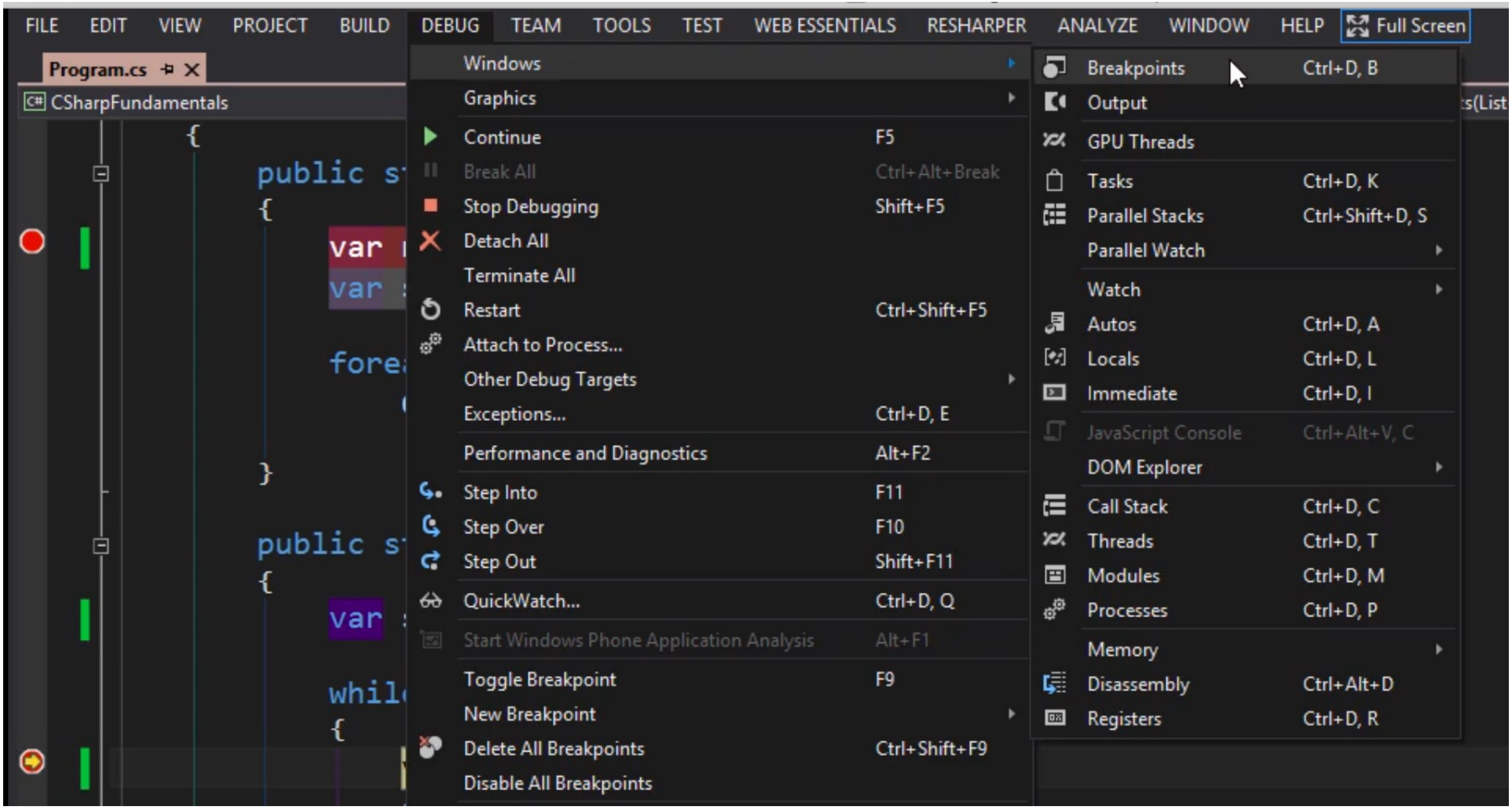

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZ
Program.cs
CSharpFundamentals
CSharpFundamentals.Program
public static int GetSmallest(List<int> list)
{
// Assume the first number is the smallest
var min = list[0];
for (var i = 1; i < list.Count; i++)
{
if (list[i] < min)
min = list[i];
}
return min;
}
}
110 %
Locals
Name
Value
list
Count = 6
i
1
min
1
Breakpoints Locals Autos
Watch 1 Call Stack Breakpoints

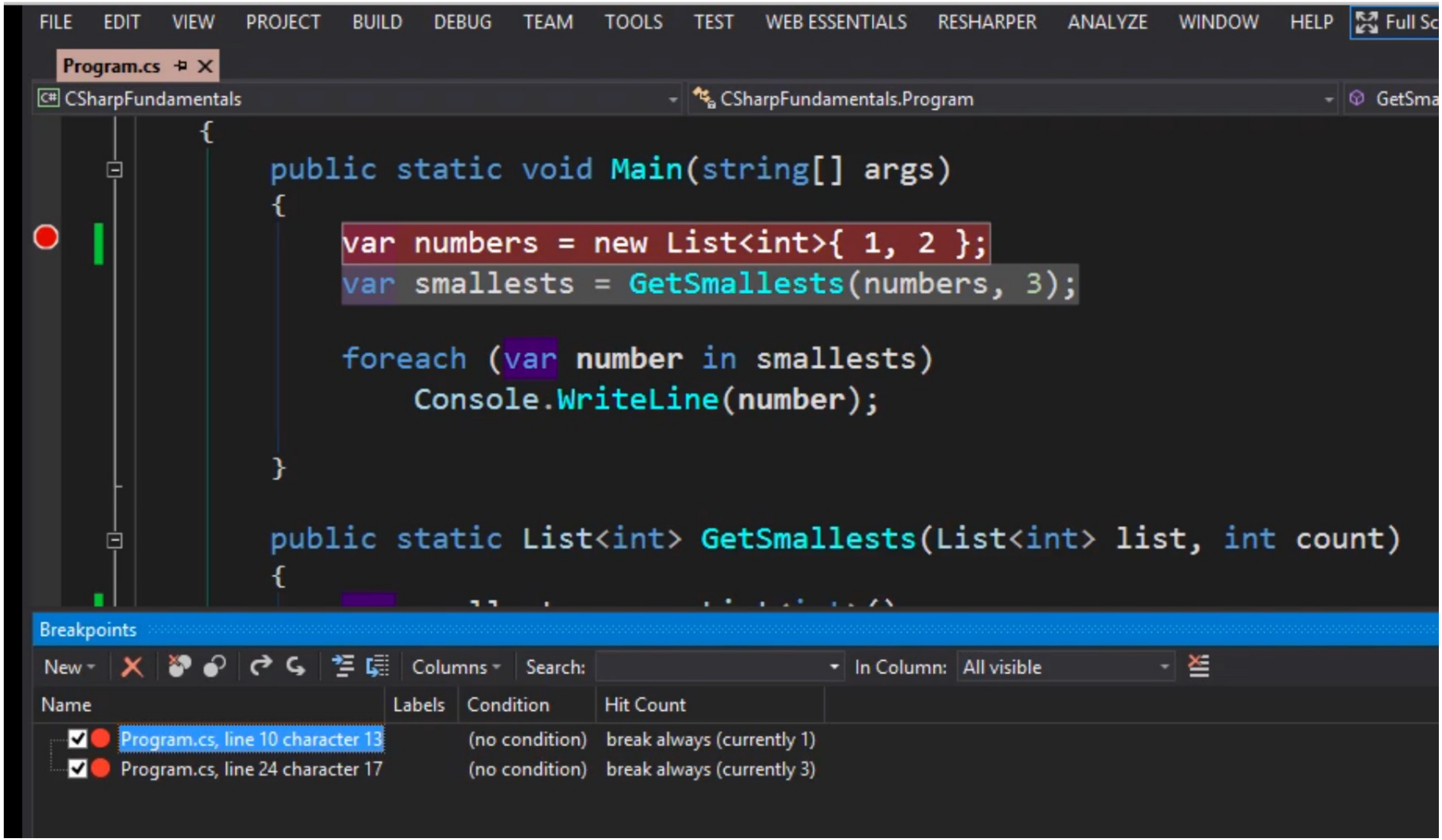
Edge Cases

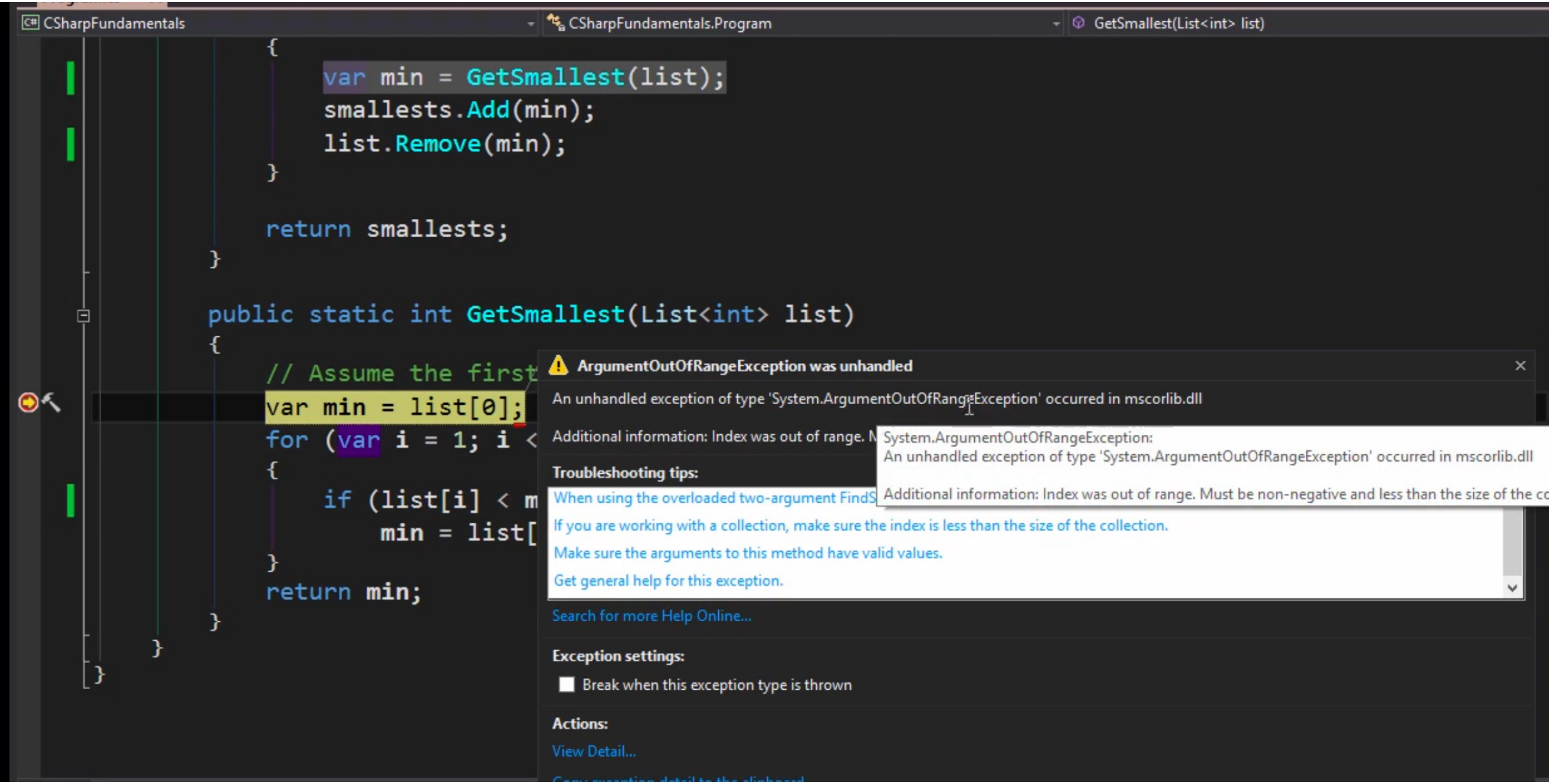
```
{
    public static void Main(string[] args)
    {
        var numbers = new List<int>{ 1, 2 };
        var smallests = GetSmallests(numbers, 3);

        foreach (var number in smallests)
            Console.WriteLine(number);

    }

    public static List<int> GetSmallests(List<int> list, int count)
    {
        var smallests = new List<int>();

        while (smallests.Count < count)
        {
            var min = GetSmallest(list);
            smallests.Add(min);
            list.Remove(min);
        }

        return smallests;
    }
```


C:\Windows\system32\cmd.exe
Unhandled Exception: System.ArgumentOutOfRangeException: Index was out of range.
Must be non-negative and less than the size of the collection.
Parameter name: index
at System.ThrowHelper.ThrowArgumentOutOfRangeException()
at System.Collections.Generic.List`1.get_Item(Int32 index)
at CSharpFundamentals.Program.
undamentals\CSharpFundamentals\Pr
at CSharpFundamentals.Program.
ojects\CSharpFundamentals\CSharpF
at CSharpFundamentals.Program.
entals\CSharpFundamentals\Program
CSharpFundamentals
CSharpFundamentals has stopped working
Windows can check online for a solution to the problem.
Check online for a solution and close the program
Close the program
Debug the program
View problem details


FILE
EDIT
VIEW
PROJECT
BUILD
DEBUG
TEAM
TOOLS
TEST
WEB ESSENTIALS
RESHARPER
Program.cs
CSharpFundamentals
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+D, E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Ctrl+D, Q
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Disable All Breakpoints
Clear All DataTips
Export DataTips ...
Import DataTips ...
Save Dump As...
Options and Settings...
CSharpFundamentals Properties...
110 %


FILE
EDIT
VIEW
PROJECT
BUILD
DEBUG
TEAM
TOOLS
TEST
WEB ESSENTIALS
RESHARPER
ANALYZE
WINDOW
HELP
Full Screen
Program.cs
CSharpFundamentals
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Attach to Process...
Other Debug Targets
Exceptions... Ctrl+D, E
Performance and Diagnostics Alt+F2
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Ctrl+D, Q
Start Windows Phone Application Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Disable All Breakpoints
Breakpoints Ctrl+D, B
Output
GPU Threads
Tasks Ctrl+D, K
Parallel Stacks Ctrl+Shift+D, S
Parallel Watch
Watch
Autos Ctrl+D, A
Locals Ctrl+D, L
Immediate Ctrl+D, I
JavaScript Console Ctrl+Alt+V, C
DOM Explorer
Call Stack Ctrl+D, C
Threads Ctrl+D, T
Modules Ctrl+D, M
Processes Ctrl+D, P
Memory
Disassembly Ctrl+Alt+D
Registers Ctrl+D, R


FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS RESHARPER ANALYZE WINDOW HELP
Program.cs
CSharpFundamentals
CSharpFundamentals.Program
{
public static void Main(string[] args)
{
var numbers = new List<int>{ 1, 2 };
var smallests = GetSmallests(numbers, 3);
foreach (var number in smallests)
Console.WriteLine(number);
}
public static List<int> GetSmallests(List<int> list, int count)
{
Breakpoints
New
Columns
Search:
In Column:
All visible
Name
Labels
Condition
Hit Count
Program.cs, line 10 character 13
(no condition)
break always (currently 1)
Program.cs, line 24 character 17
(no condition)
break always (currently 3)

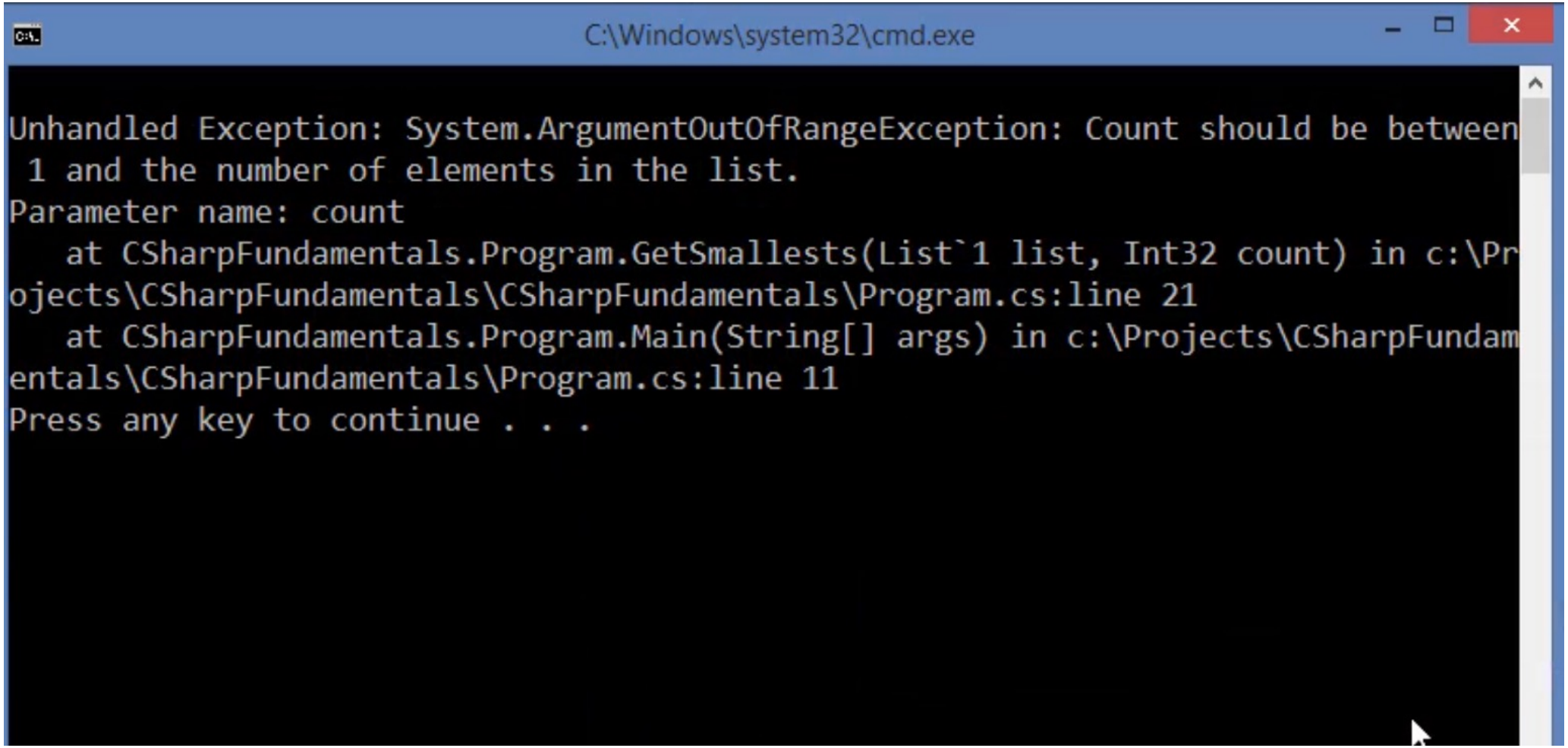

CSharpFundamentals
CSharpFundamentals.Program
GetSmallest(List<int> list)
{
var min = GetSmallest(list);
smallests.Add(min);
list.Remove(min);
}
return smallests;
}
public static int GetSmallest(List<int> list)
{
// Assume the first
var min = list[0];
for (var i = 1; i <
{
if (list[i] < m
min = list[
}
return min;
}
}
}
ArgumentOutOfRangeException was unhandled
An unhandled exception of type 'System.ArgumentOutOfRangeException' occurred in mscorlib.dll
Additional information: Index was out of range. M
System.ArgumentOutOfRangeException:
An unhandled exception of type 'System.ArgumentOutOfRangeException' occurred in mscorlib.dll
Additional information: Index was out of range. Must be non-negative and less than the size of the co
Troubleshooting tips:
When using the overloaded two-argument FindS
If you are working with a collection, make sure the index is less than the size of the collection.
Make sure the arguments to this method have valid values.
Get general help for this exception.
Search for more Help Online...
Exception settings:
Break when this exception type is thrown
Actions:
View Detail...

```
}

public static List<int> GetSmallests(List<int> list, int count)
{
    var smallests = new List<int>();

    while (smallests.Count < count)
    {
        var min = GetSmallest(list);
        smallests.Add(min);
        list.Remove(min);
    }

    return smallests;
}

public static int GetSmallest(List<int> list)
{
    // Assume the first number is the smallest
    var min = list[0];
    for (var i = 1; i < list.Count; i++)
    {
        if (list[i] < min)
            min = list[i];
    }
```

```
public static List<int> GetSmallests(List<int> list, int count)
{
    if (count > list.Count)
        throw new ArgumentOutOfRangeException("count");

    var buffer = new List<int>(list);
    var smallests = new List<int>();

    while (smallests.Count < count)
    {
        var min = GetSmallest(buffer);
        smallests.Add(min);
        buffer.Remove(min);
    }

    return smallests;
}

public static int GetSmallest(List<int> list)
{
    // Assume the first number is the smallest
    var min = list[0];
    for (var i = 1; i < list.Count; i++)
    {
```

```
public static List<int> GetSmallests(List<int> list, int count)
{
    if (count > list.Count)
        throw new ArgumentOutOfRangeException("count", "Count cannot be greater than the numb

    var buffer = new List<int>(list);
    var smallests = new List<int>();

    while (smallests.Count < count)
    {
        var min = GetSmallest(buffer);
        smallests.Add(min);
        buffer.Remove(min);
    }

    return smallests;
}

public static int GetSmallest(List<int> list)
{
    // Assume the first number is the smallest
    var min = list[0];
    for (var i = 1; i < list.Count; i++)
```

```
public static void Main(string[] args)
{
    var numbers = new List<int>{ 1, 2 };
    var smallests = GetSmallests(numbers, -1);

    foreach (var number in smallests)
        Console.WriteLine(number);

}

public static List<int> GetSmallests(List<int> list, int count)
{
    if (count > list.Count || count <= 0)
        throw new ArgumentOutOfRangeException("count", "Count should be between 1 and the n

    var buffer = new List<int>(list);
    var smallests = new List<int>();

    while (smallests.Count < count)
    {
        var min = GetSmallest(buffer);
        smallests.Add(min);
        buffer.Remove(min);
    }
```


C:\Windows\system32\cmd.exe
Unhandled Exception: System.ArgumentOutOfRangeException: Count should be between
1 and the number of elements in the list.
Parameter name: count
at CSharpFundamentals.Program.GetSmallests(List`1 list, Int32 count) in c:\Pr
ojects\CSharpFundamentals\CSharpFundamentals\Program.cs:line 21
at CSharpFundamentals.Program.Main(String[] args) in c:\Projects\CSharpFundam
entals\CSharpFundamentals\Program.cs:line 11
Press any key to continue . . .