Debugging

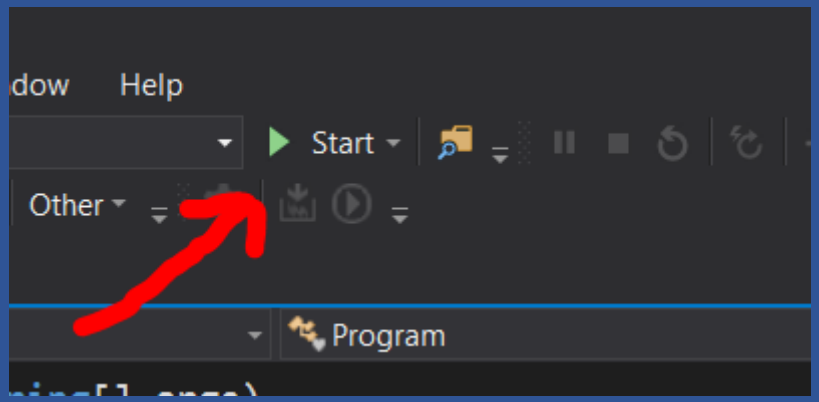
What is Debugging?

- Finding and resolving defects
- Interactive debugging
- Control flow analysis
- Unit testing
- Integration testing
- Log file analysis
- Monitoring at the application

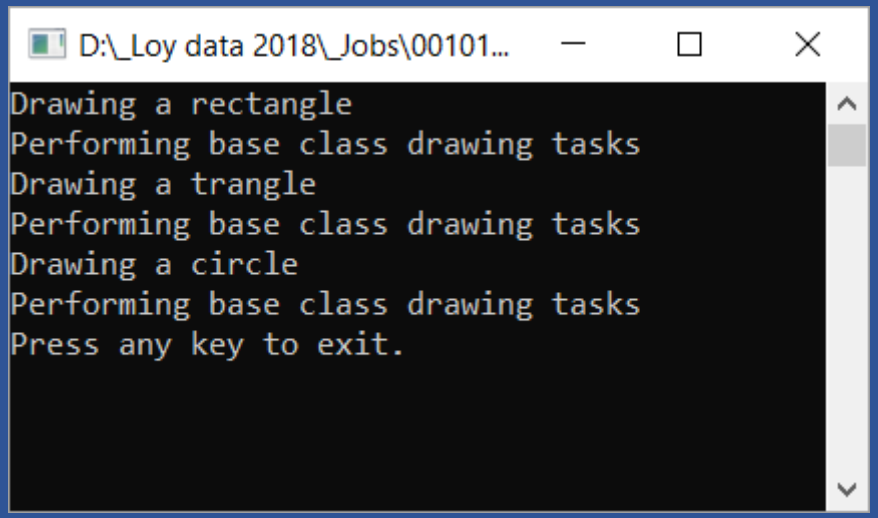
In this session

- Start the debugger and hit breakpoints.
- Learn commands to step through code in the debugger
- Inspect variables in data tips and debugger windows
- Examine the call stack

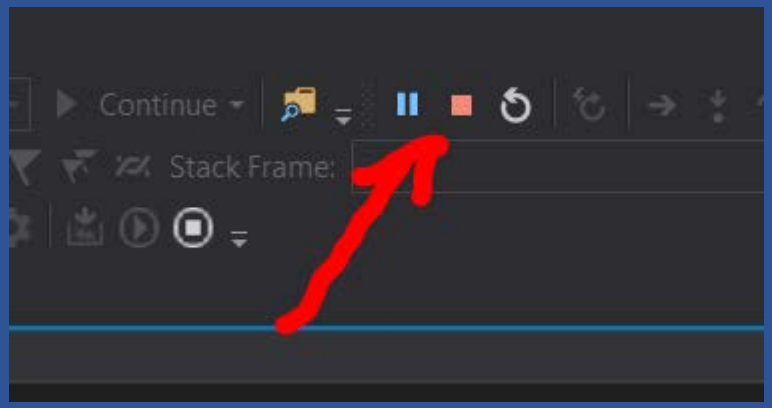
- Create Consol app “debug1”
- Copy code from “CSharp2/Source Code/00135 Debugging.cs”
- Press Start Debugging



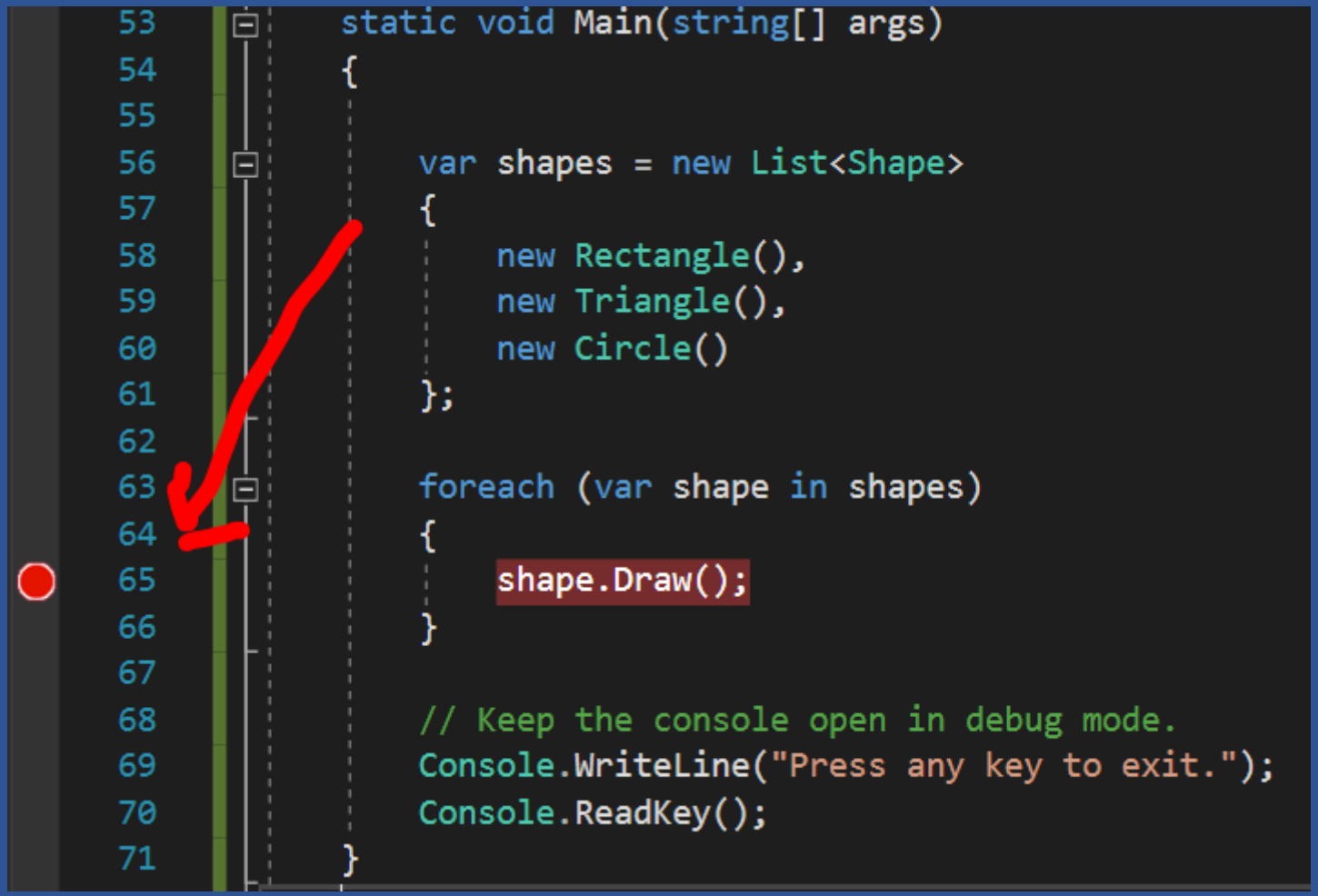
- Consol appears



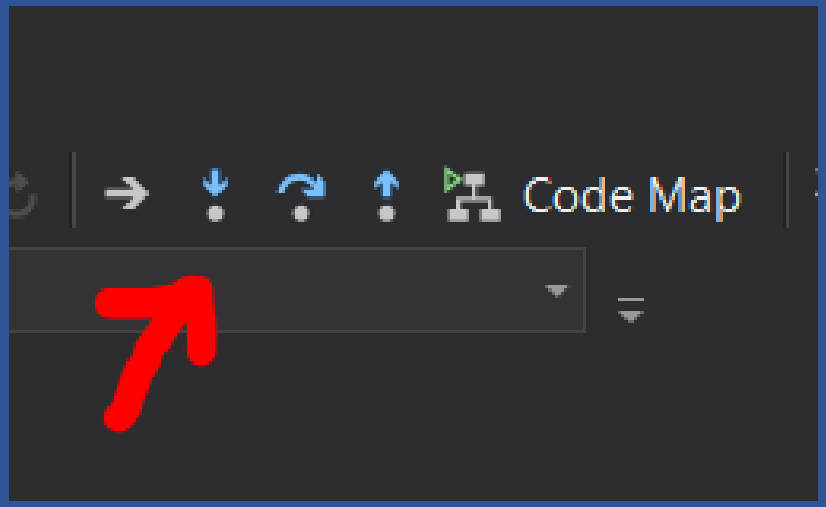
- Press “Stop” to stop debugging



- Set “break point” here

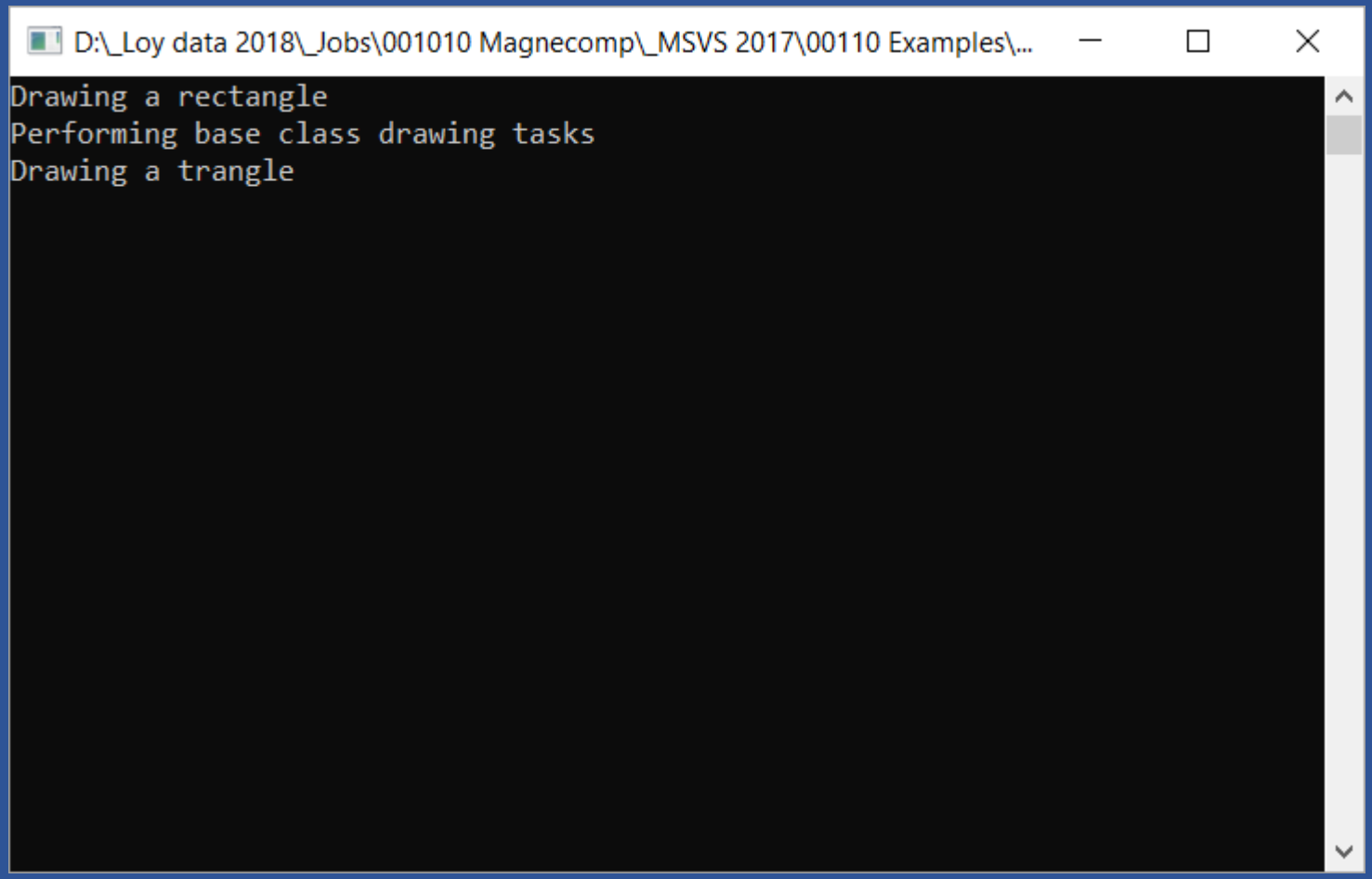


- Navigate code in the debugger using step commands

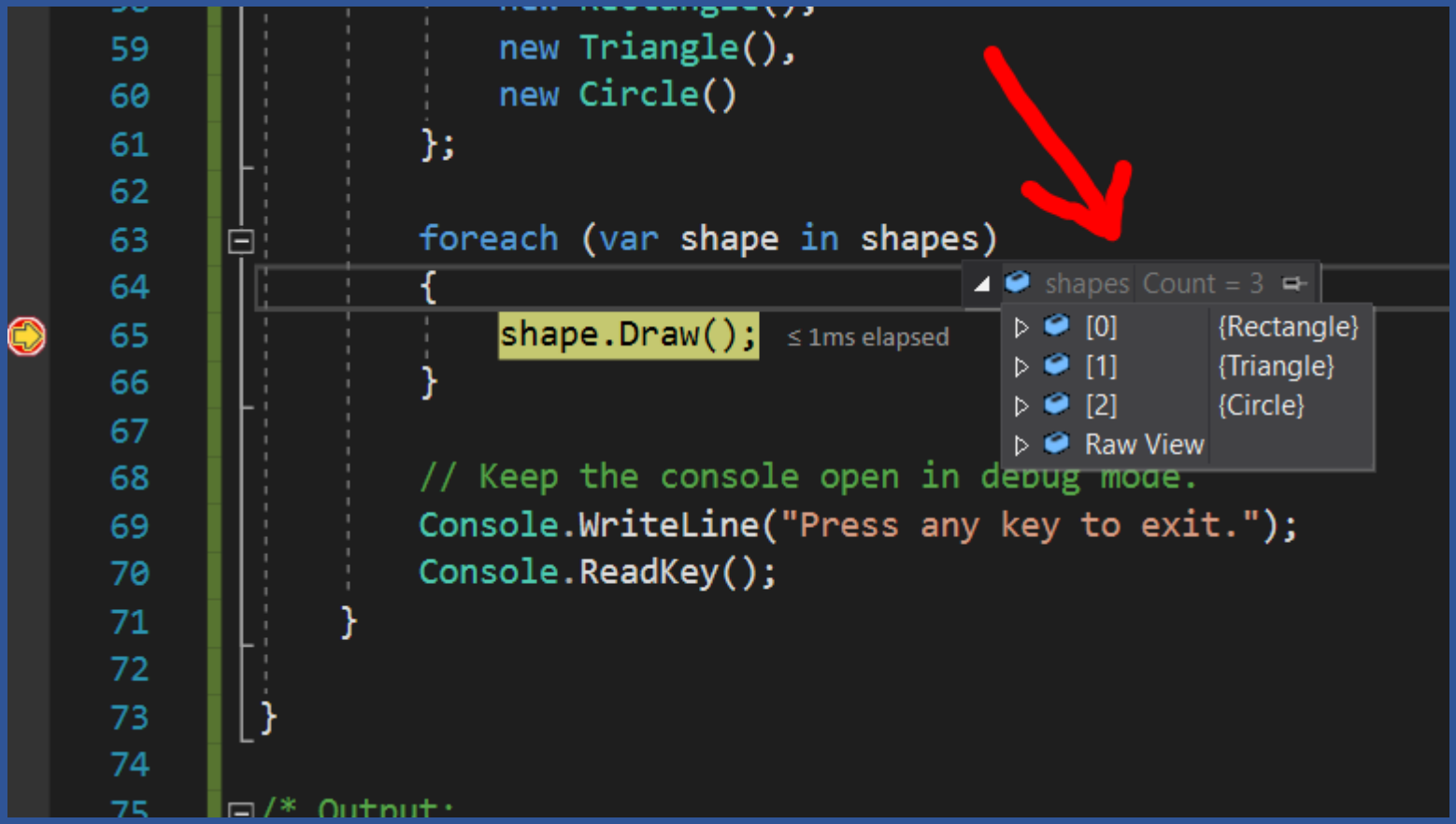


- Step into
- Step over
- Stop out

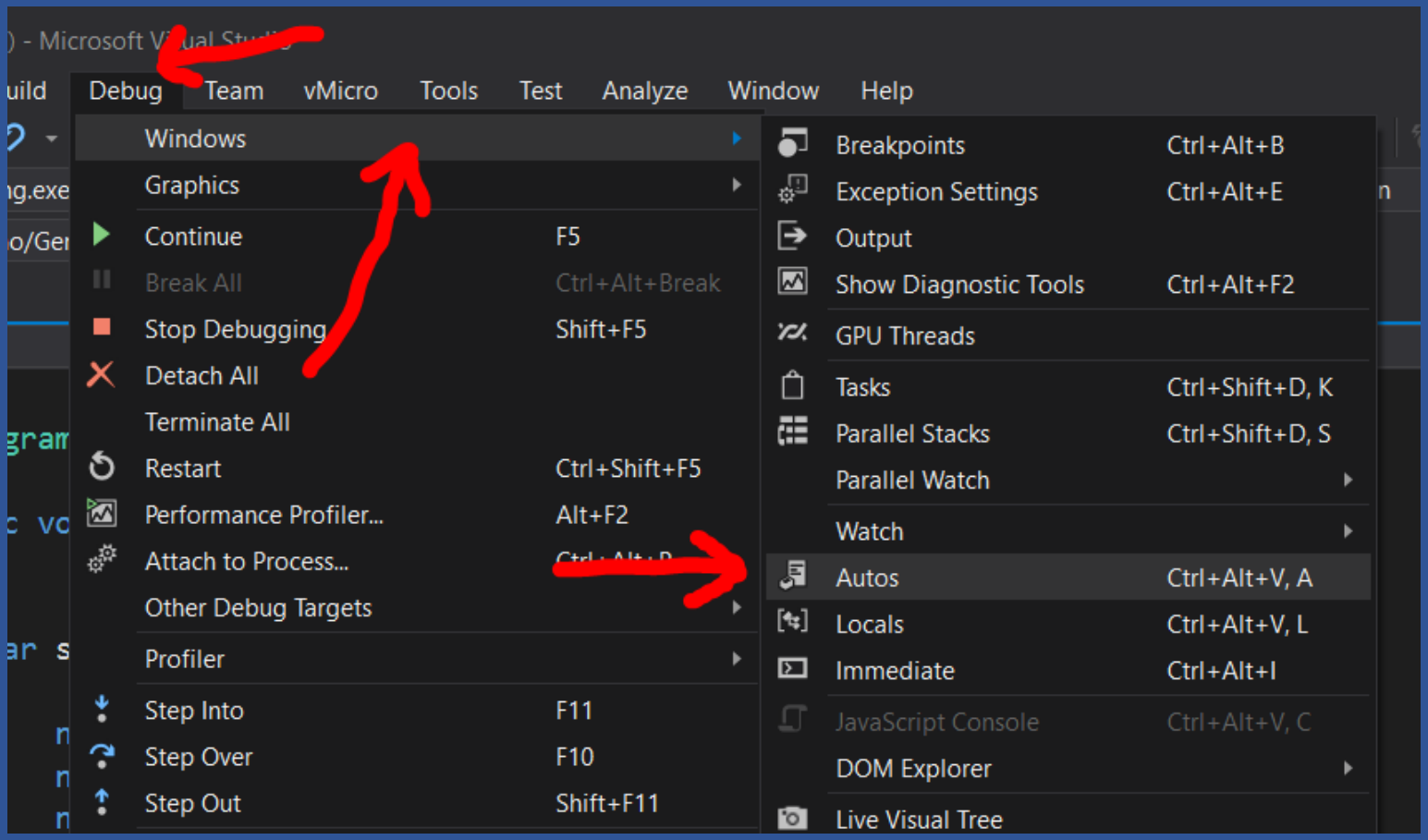
- Watch the console while stop



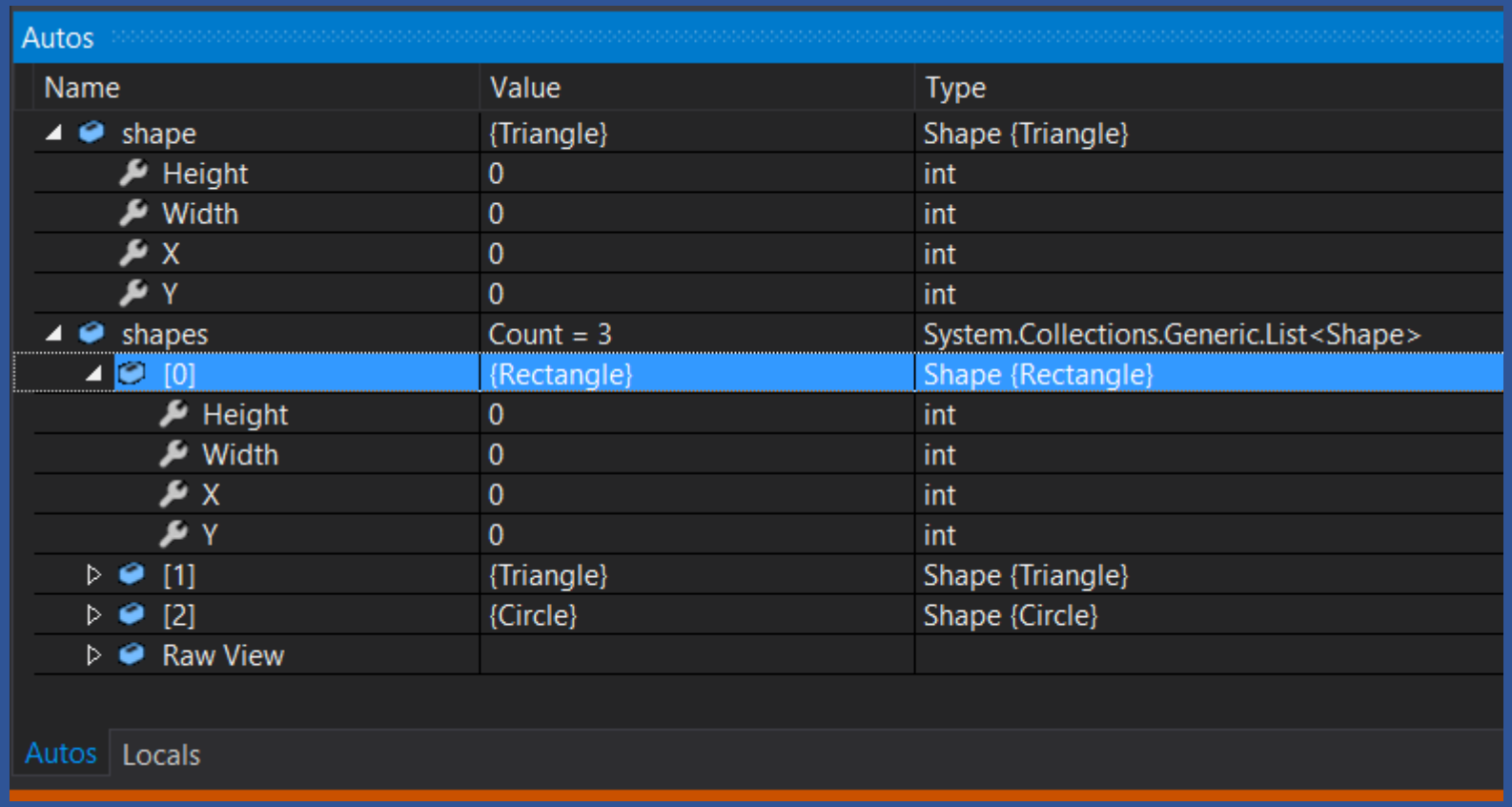
- Inspect variables with data tips



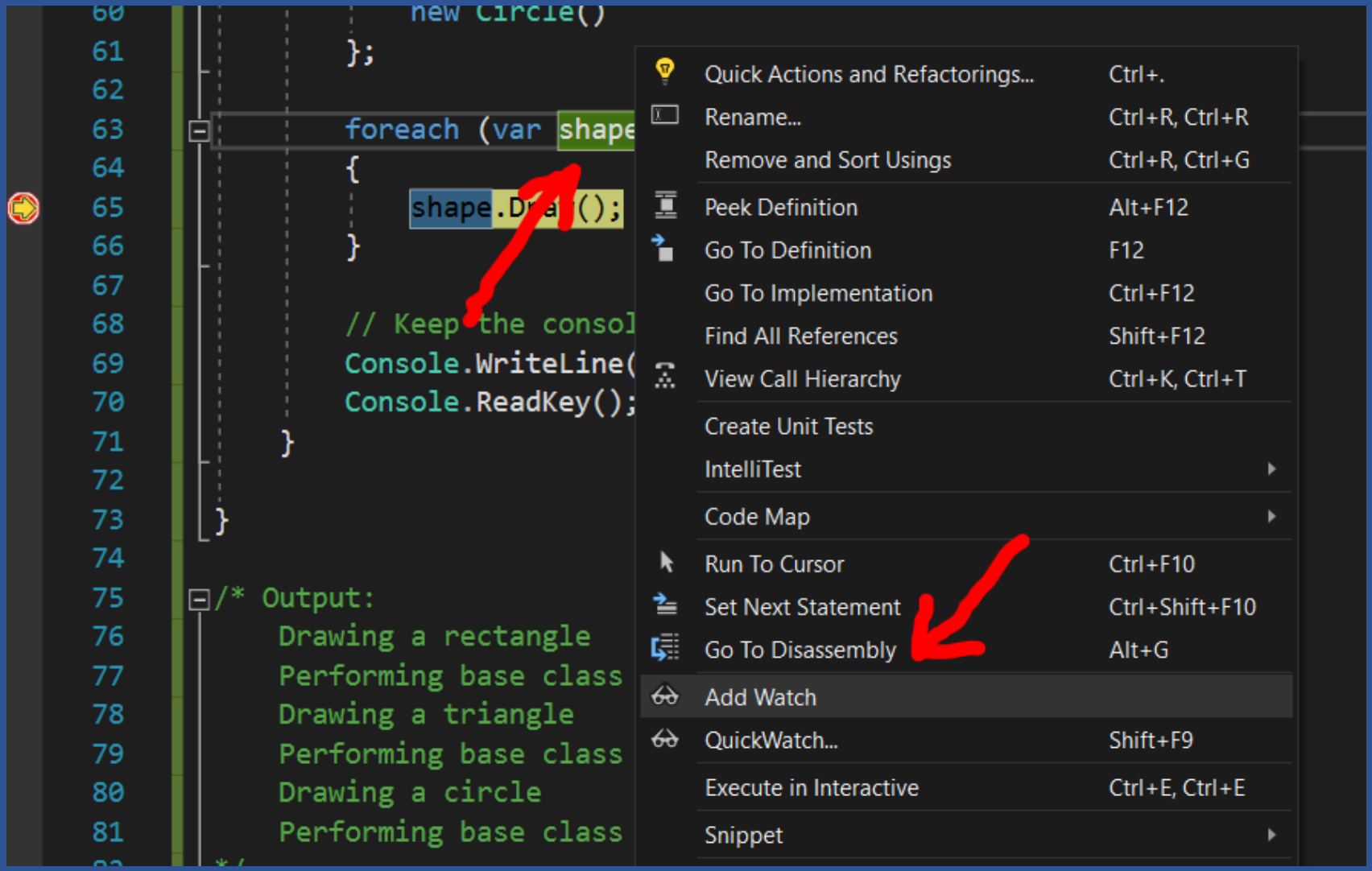
- Inspect variables with the Autos and Locals windows



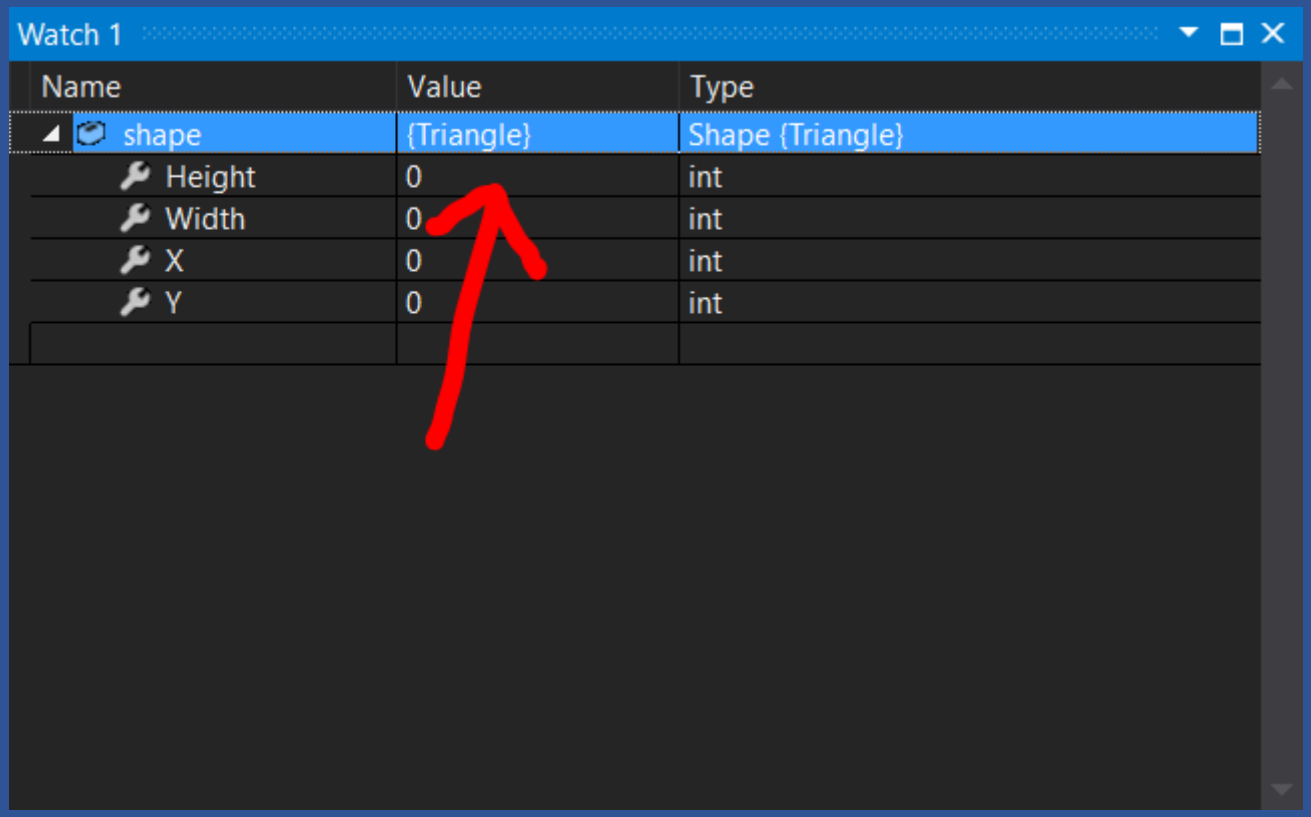
- Watch Autos windows while step



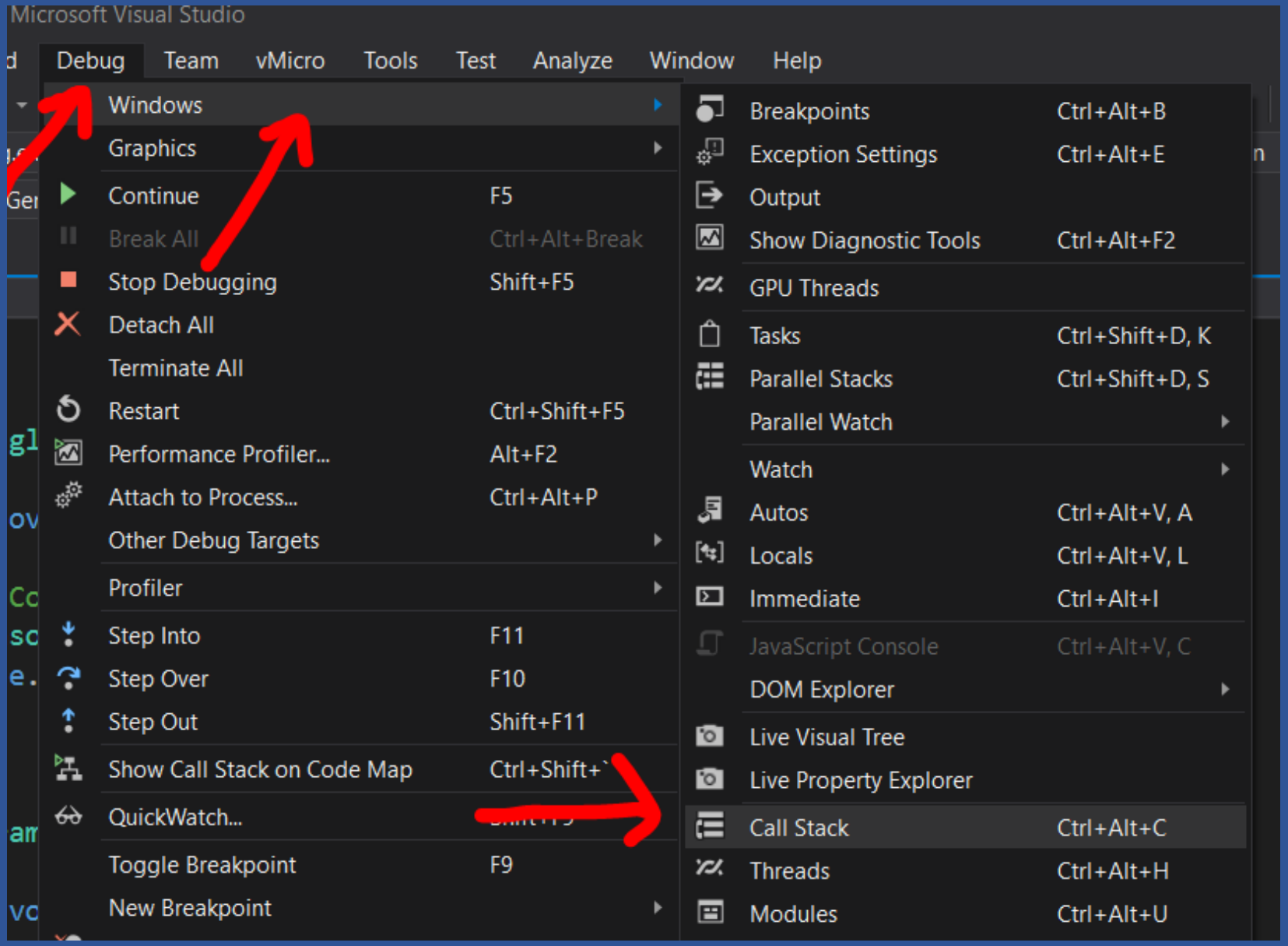
- Add watch


new Circle()
};
foreach (var shape
{
shape.Dra();
}
// Keep the consol
Console.WriteLine(
Console.ReadKey();
}
}
/* Output:
Drawing a rectangle
Performing base class
Drawing a triangle
Performing base class
Drawing a circle
Performing base class
Quick Actions and Refactorings... Ctrl+.
Rename... Ctrl+R, Ctrl+R
Remove and Sort Usings Ctrl+R, Ctrl+G
Peek Definition Alt+F12
Go To Definition F12
Go To Implementation Ctrl+F12
Find All References Shift+F12
View Call Hierarchy Ctrl+K, Ctrl+T
Create Unit Tests
IntelliTest
Code Map
Run To Cursor Ctrl+F10
Set Next Statement Ctrl+Shift+F10
Go To Disassembly Alt+G
Add Watch
QuickWatch... Shift+F9
Execute in Interactive Ctrl+E, Ctrl+E
Snippet

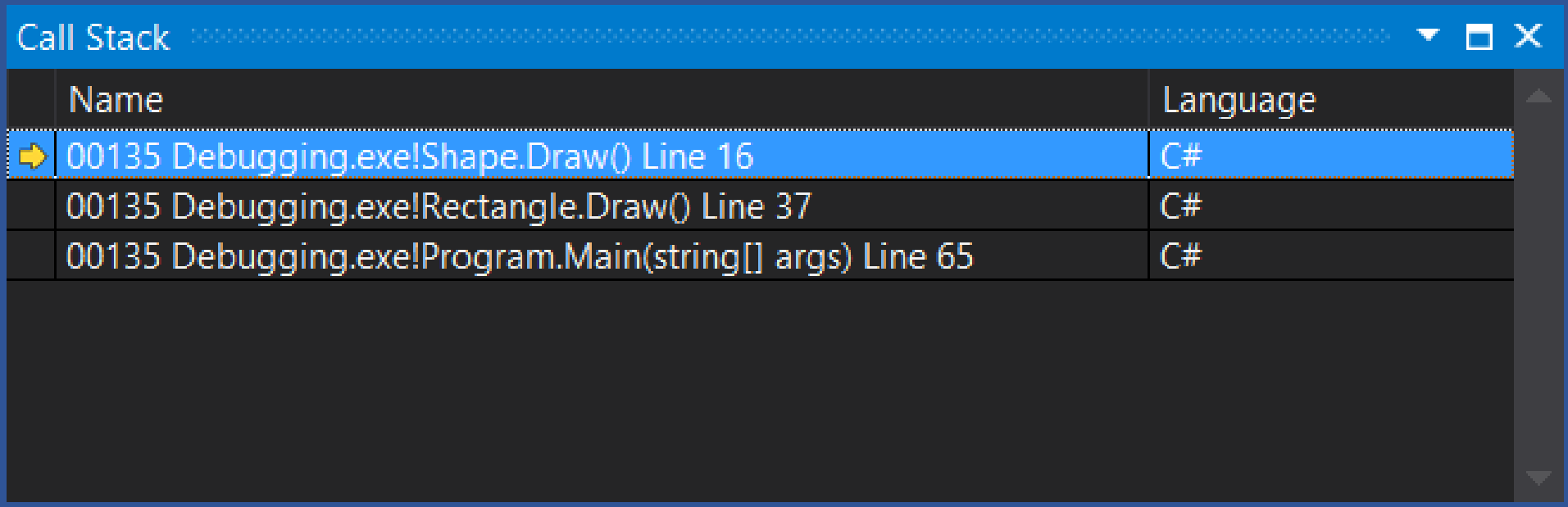
- Watch the value change while step



- Examine the call stack
  - The Call Stack window shows the order in which methods and functions are getting called.
  - The top line shows the current function


Microsoft Visual Studio
Debug
Team
vMicro
Tools
Test
Analyze
Window
Help
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Performance Profiler... Alt+F2
Attach to Process... Ctrl+Alt+P
Other Debug Targets
Profiler
Step Into F11
Step Over F10
Step Out Shift+F11
Show Call Stack on Code Map Ctrl+Shift+`
QuickWatch...
Toggle Breakpoint F9
New Breakpoint
Breakpoints Ctrl+Alt+B
Exception Settings Ctrl+Alt+E
Output
Show Diagnostic Tools Ctrl+Alt+F2
GPU Threads
Tasks Ctrl+Shift+D, K
Parallel Stacks Ctrl+Shift+D, S
Parallel Watch
Watch
Autos Ctrl+Alt+V, A
Locals Ctrl+Alt+V, L
Immediate Ctrl+Alt+I
JavaScript Console Ctrl+Alt+V, C
DOM Explorer
Live Visual Tree
Live Property Explorer
Call Stack Ctrl+Alt+C
Threads Ctrl+Alt+H
Modules Ctrl+Alt+U

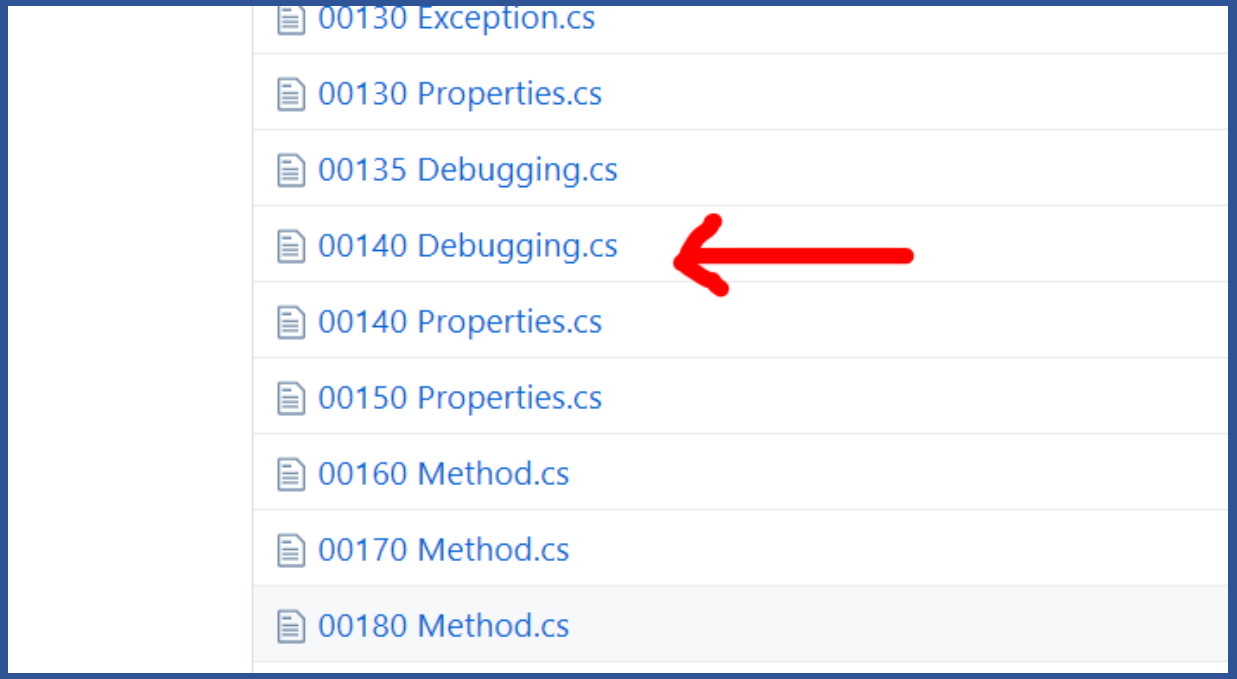
- Examine the Call Stack while step



- Double-click line to see the source code

Find bug using debugging tools

- Create new consol app “debug2”
- Copy code from “CSharp2/Source Code/00140 Debugging.cs”

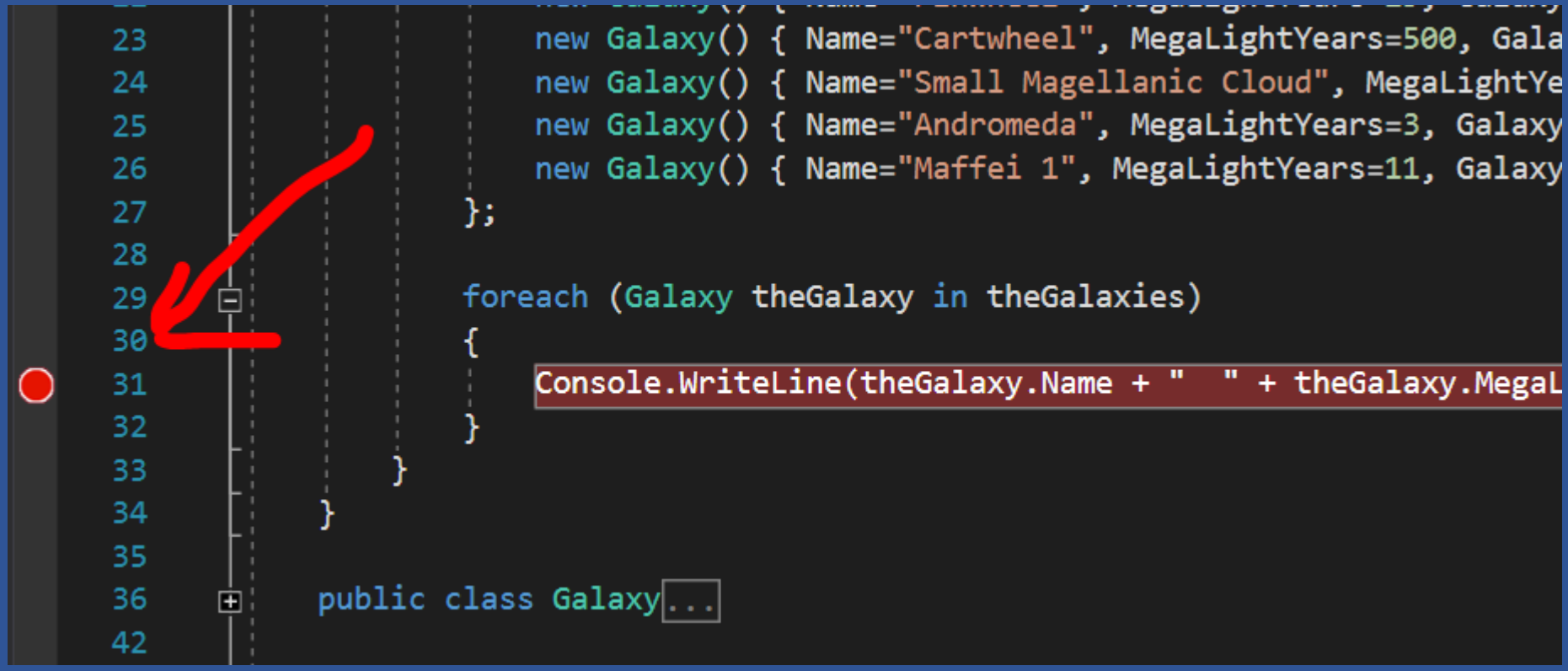


- Run program
- Examine the result

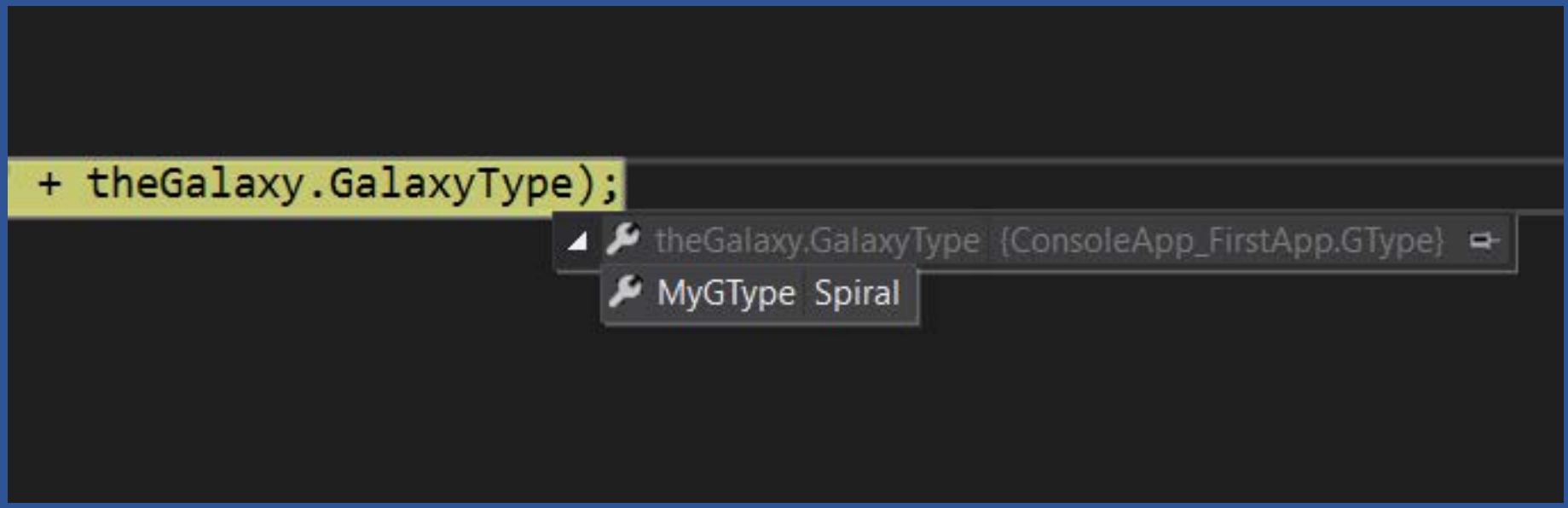
```
/*
 ------ actual output ---------
Welcome to Galaxy News!
Tadpole  400,  ConsoleApp_FirstApp.GType
Pinwheel  25,  ConsoleApp_FirstApp.GType
Cartwheel  500,  ConsoleApp_FirstApp.GType
Small Magellanic Cloud  0.2,  ConsoleApp_FirstApp.GType
Andromeda  3,  ConsoleApp_FirstApp.GType
Maffei 1  11,  ConsoleApp_FirstApp.GType

----- expect to see this --------
Tadpole  400,  Spiral
Pinwheel  25,  Spiral
Cartwheel, 500,  Lenticular
Small Magellanic Cloud .2,  Irregular
Andromeda  3,  Spiral
Maffei 1,  Elliptical
```

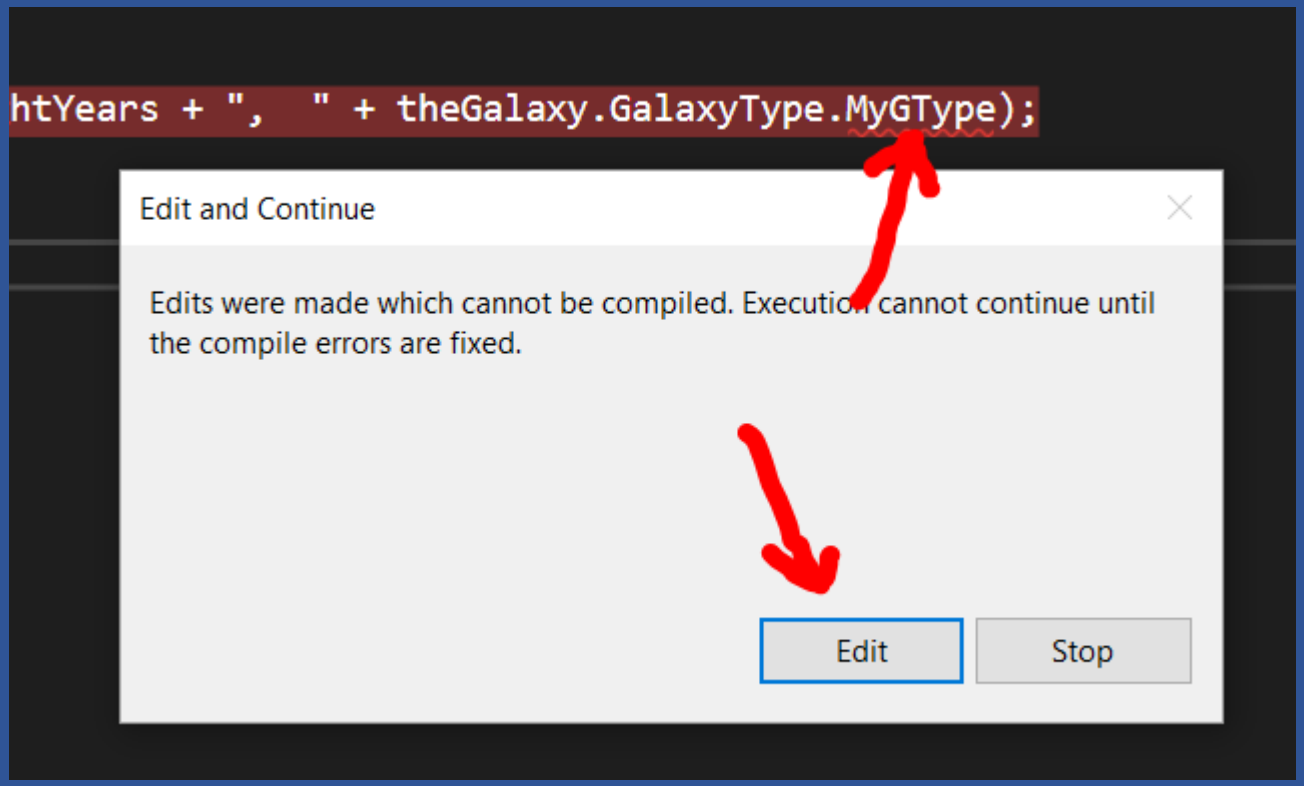
- Set break point here



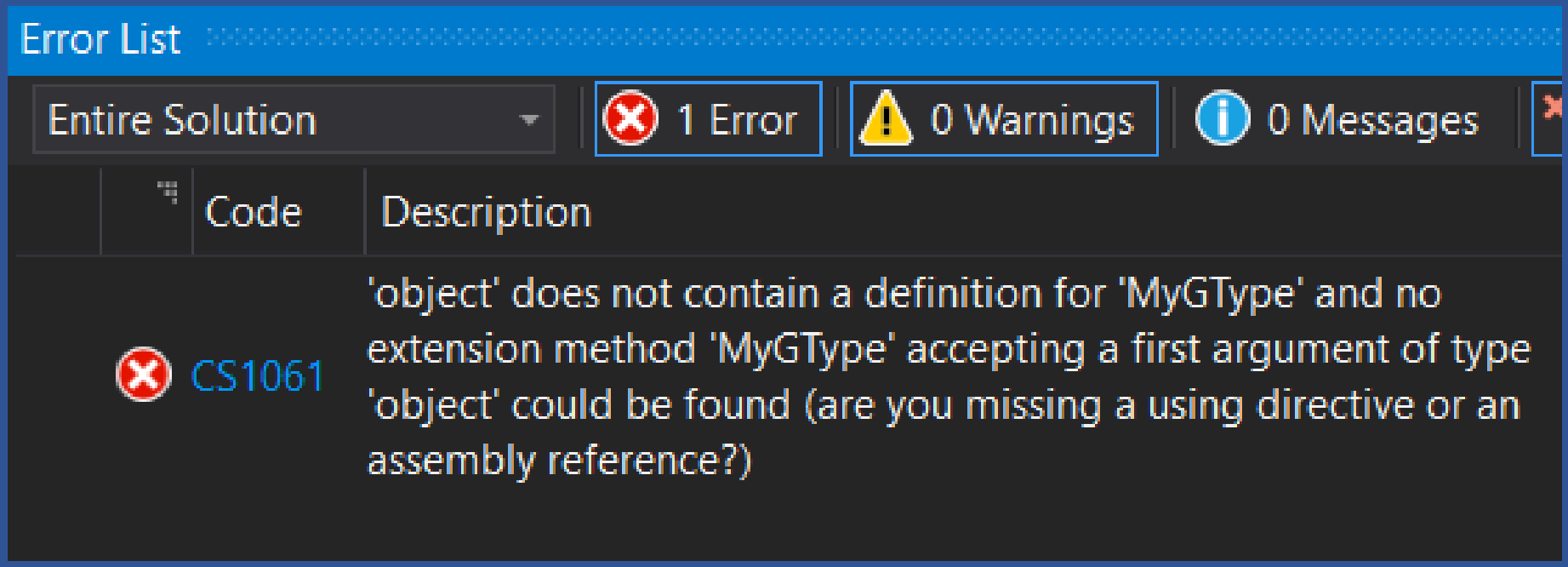
- Run program
- Hover mouse over GalaxyType



- Change theGalaxy.GalaxyType to theGalaxy.GalaxyType.MyGType



- Look at the error message



- the type appears to be an object of type object instead of an object of type GType

- Looking at galaxy type, you find the GalaxyType property of the Galaxy class is specified as object instead of GType

```
61                  default:
62                      break;
63                  }
64              }
65              public object MyGType { get; set; }
66              private enum Type { Spiral, Elliptical, Irregular, Lenticular }
67          }
68      }
```

- Change from object to GType

```
public GType GalaxyType { get; set; }
```

Exercise

- Stop debug
- Start debug
- When break, Examine variable Galaxy theGalaxy using;
  - Data tip
  - Autos / Local window
  - Call Stack