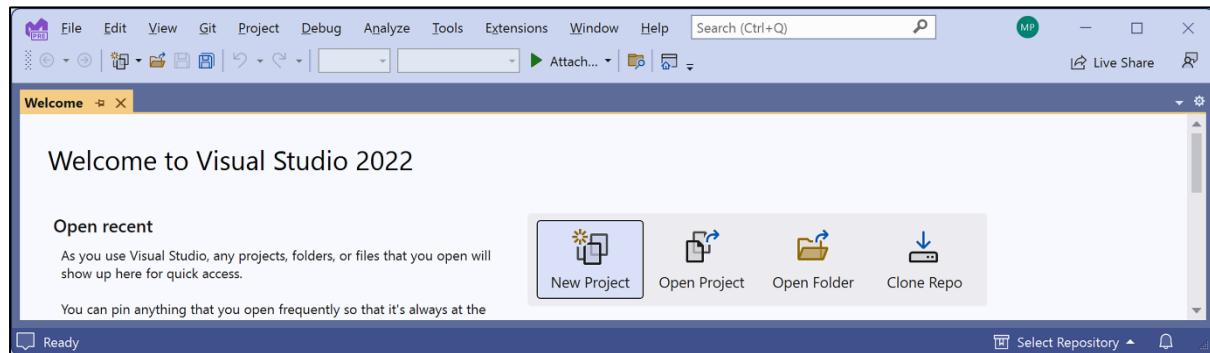
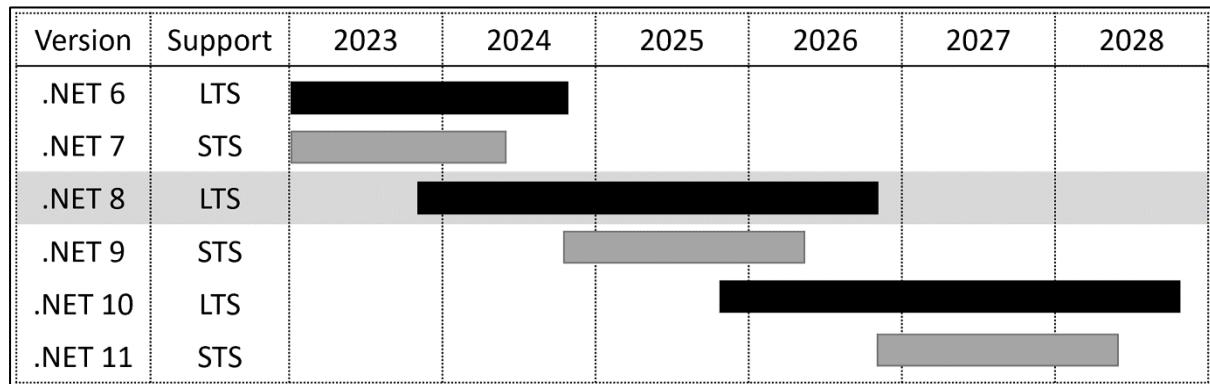
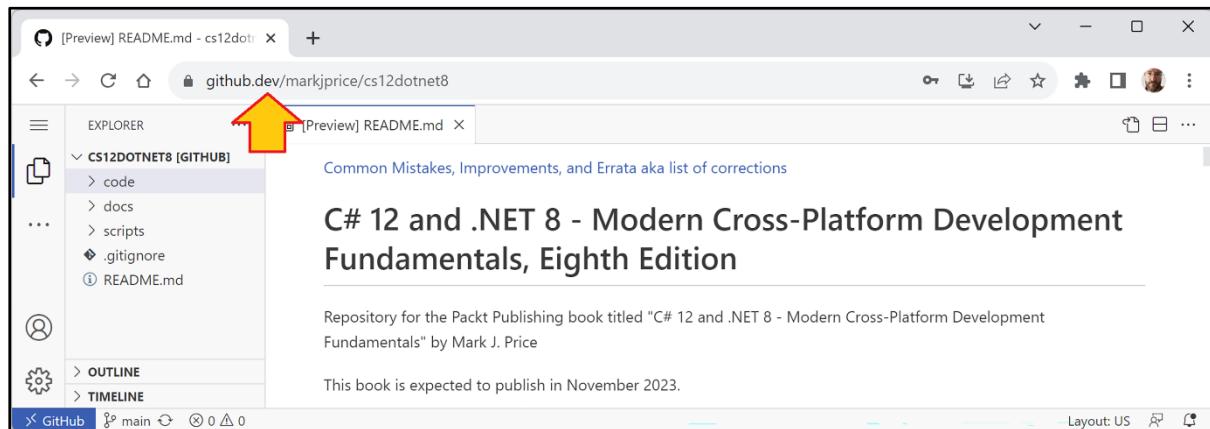
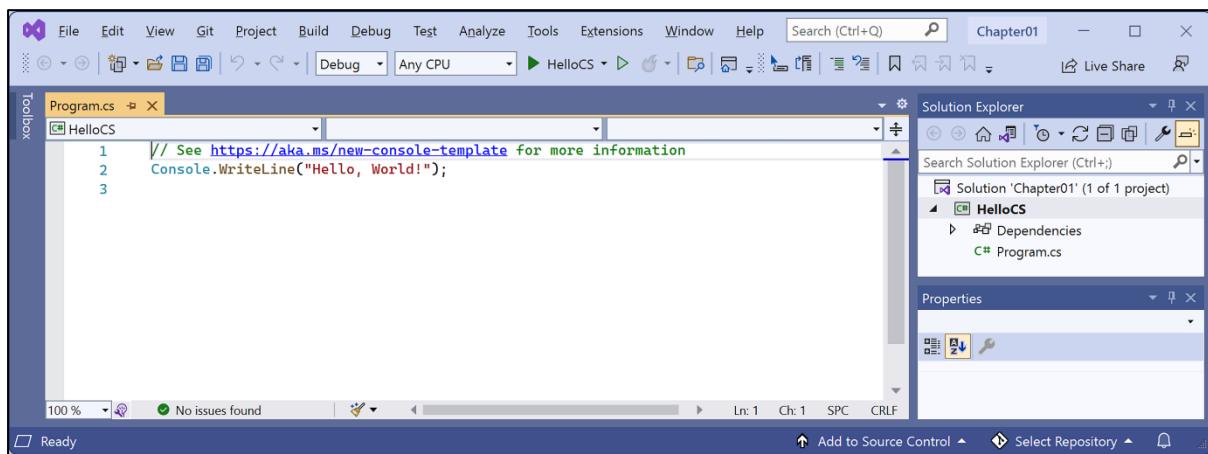
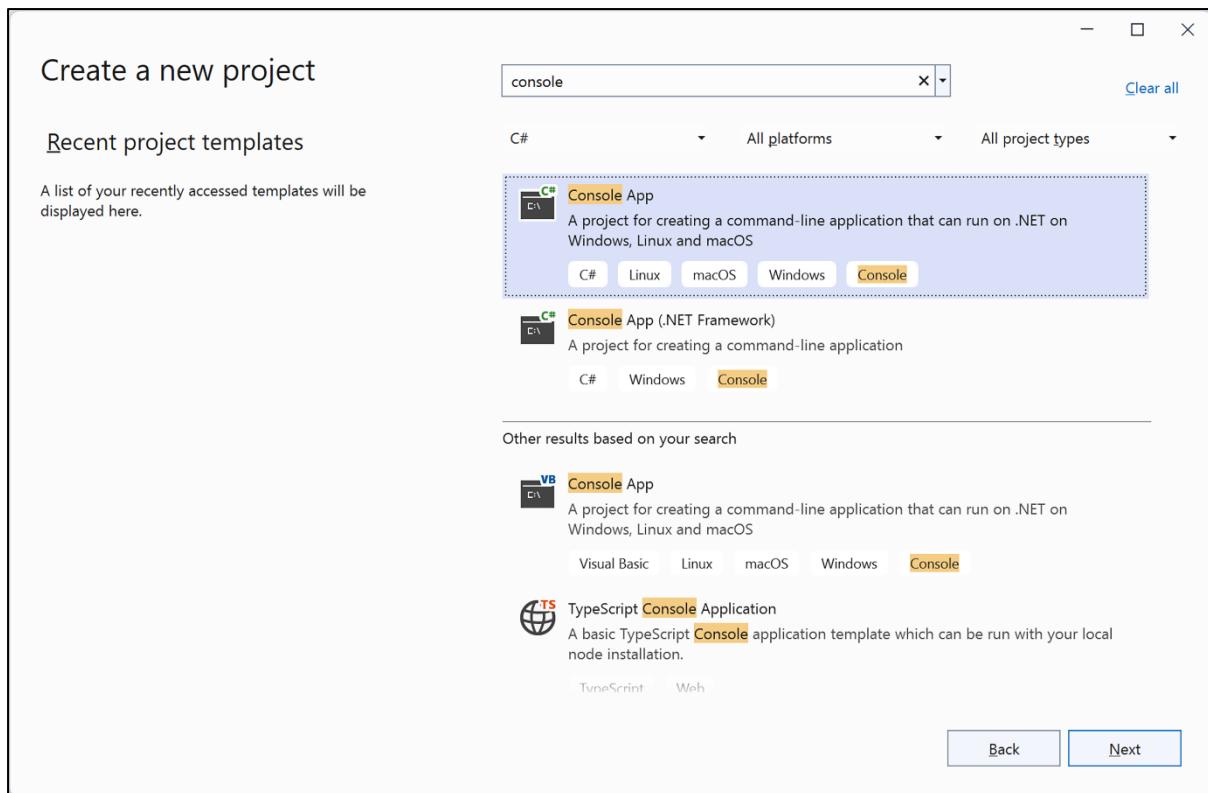
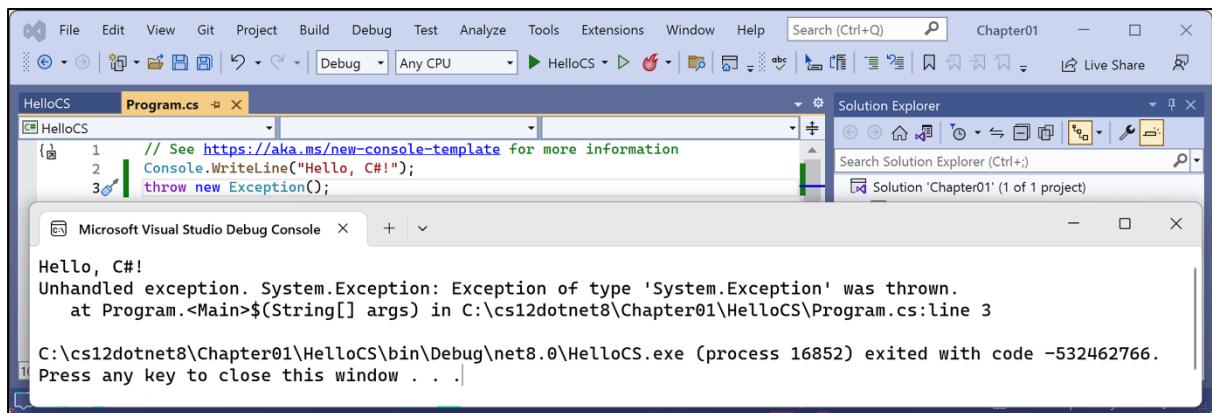
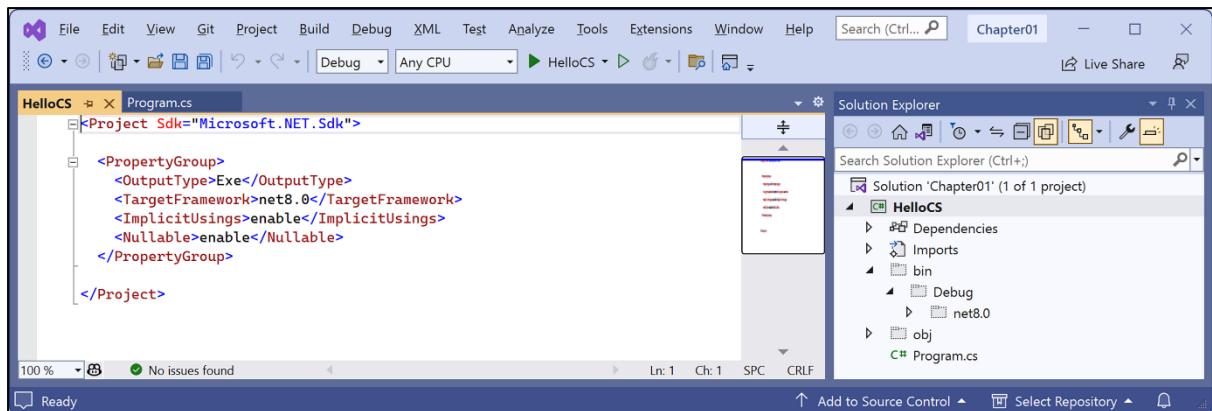
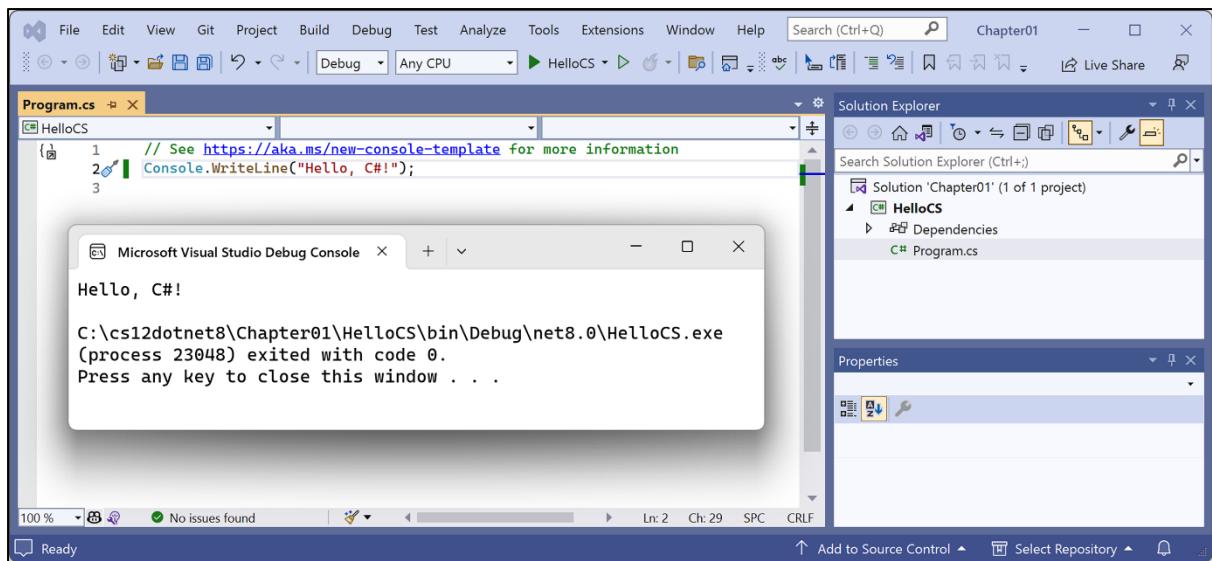


Chapter 1: Hello, C#! Welcome, .NET!







Screenshot of Microsoft Visual Studio showing the `AboutMyEnvironment` project in the Solution Explorer. The `Program.cs` file is open in the editor, displaying the following C# code:

```
1  namespace AboutMyEnvironment
2  {
3      internal class Program
4      {
5          static void Main(string[] args)
6          {
7              Console.WriteLine(Environment.CurrentDirectory);
8              Console.WriteLine(Environment.OSVersion.VersionString);
9              Console.WriteLine("Namespace: {0}", typeof(Program).Namespace);
10         }
11     }
12 }
```

The output window shows the following console output:

```
C:\cs12dotnet8\Chapter01\AboutMyEnvironment\bin\Debug\net8.0
Microsoft Windows NT 10.0.22621.0
Namespace: AboutMyEnvironment

C:\cs12dotnet8\Chapter01\AboutMyEnvironment\bin\Debug\net8.0\AboutMyEnvironment.exe (process 25296) exited with code 0.
Press any key to close this window . . .
```

Screenshot of Visual Studio Code showing the `CHAPTER01-VSCODE` project in the Explorer sidebar. The Welcome screen displays various quick start options and recent projects.

Welcome - Chapter01-vscode - Visual Studio Code

EXPLORER

- > CHAPTER01-VSCODE
- > .vscode
- > HelloCS
- > bin
- > obj
- >HelloCS.csproj
- Program.cs
- Chapter01.sln

Start

- + New File...
- Open File...
- Open Folder...
- Clone Git Repository...

Walkthroughs

- Get Started with C# Dev...
- Get Started with Jupyter...
- Get Started with WSL
- Browse & Edit Remote ...
- Getting Started with D...

Recent

- Chapter01 C:\cs12dotnet8
- HelloCS C:\cs12dotnet8\Chapter01-vscode
- AboutMyEnvironment C:\cs12dotnet8\Chapter...

Screenshot of Visual Studio Code showing the `CHAPTER01-VSCODE` project in the Explorer sidebar. The Output tab shows the following log entries:

```
Starting Open a solution...
Starting Open a solution with environment service...
Starting Clear environment...
Starting Spawn .NET server...
.NET server started and IPC established in 1841ms
Completed Spawn .NET server (1853ms)
Completed Clear environment (2165ms)
Completed Open a solution with environment service (2362ms)
Starting Restore solution...
Completed Open a solution (2399ms)
Completed Restore solution (1571ms)
```

The screenshot shows a Visual Studio Code interface with the following details:

- Solution Explorer:** Shows a project named "HelloCS" with a "Program.cs" file selected.
- Code Editor:** Displays the following C# code:

```
1 See https://aka.ms/new-console-template for more information
2 Console.WriteLine("Hello, C#!");
3
```
- Terminal:** Shows the command "dotnet run" being run, resulting in the output "Hello, C#!".
- Status Bar:** Shows "Projects: 1" and other standard status bar information.

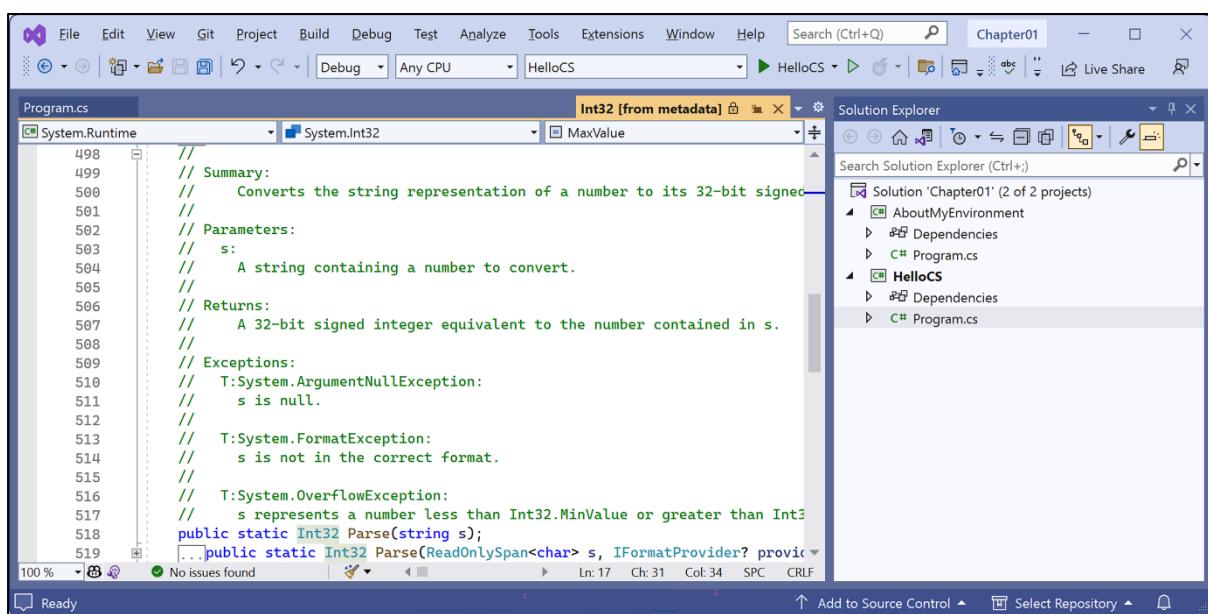
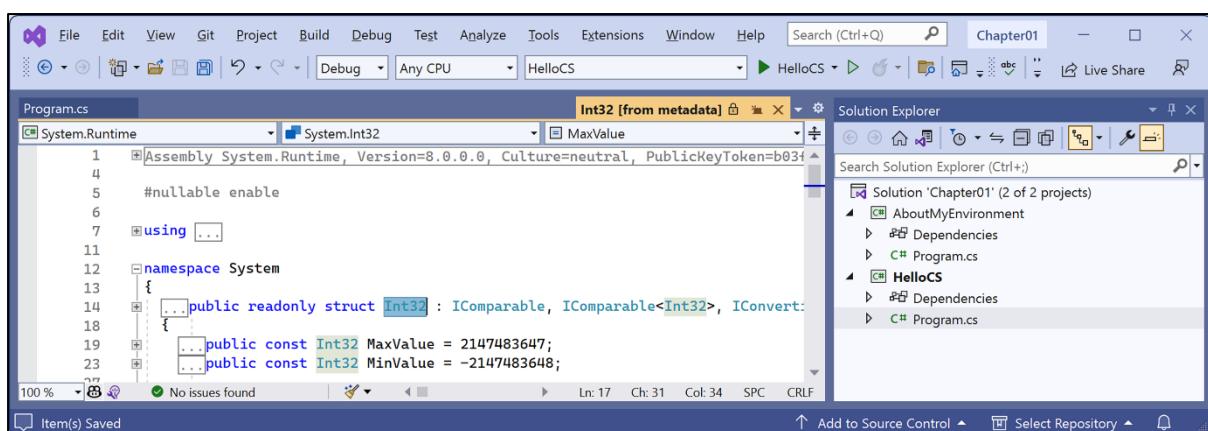
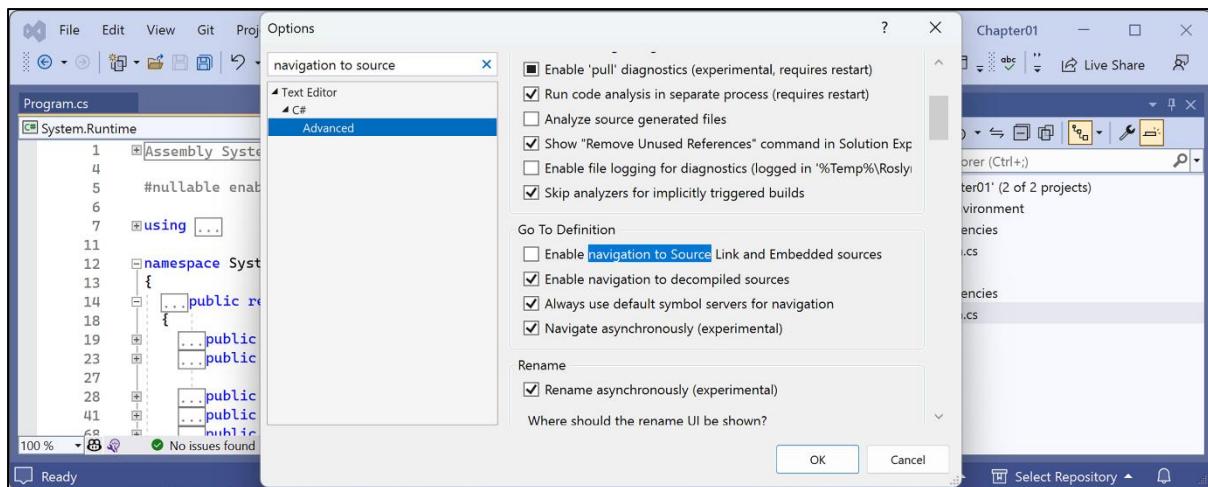
The screenshot shows a Visual Studio Code interface with the following details:

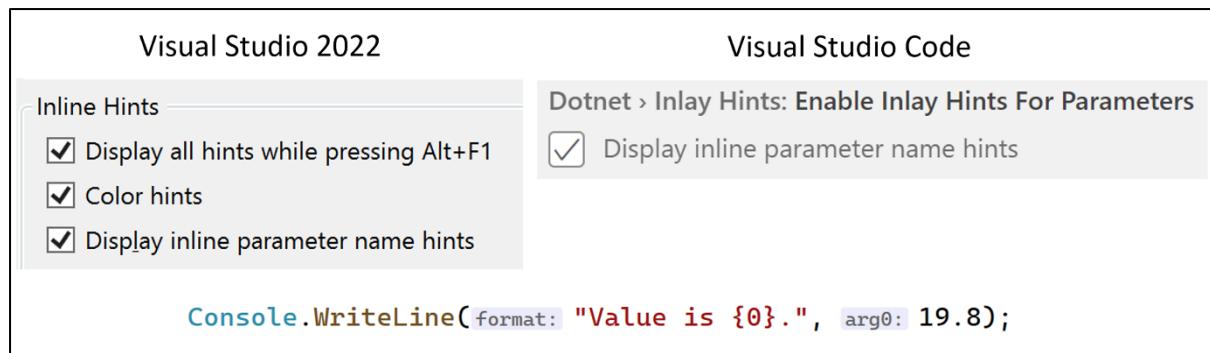
- Solution Explorer:** Shows a project named "AboutMyEnvironment" with a "Program.cs" file selected.
- Code Editor:** Displays the following C# code:

```
1 namespace AboutMyEnvironment;
2
3     1 reference
4     class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             Console.WriteLine(Environment.CurrentDirectory);
10            Console.WriteLine(Environment.OSVersion.VersionString);
11            Console.WriteLine("Namespace: {0}", typeof(Program).Namespace);
12        }
13    }
```
- Terminal:** Shows the command "dotnet run" being run, resulting in the output "C:\cs12dotnet8\Chapter01-vscode\AboutMyEnvironment" followed by system information and the namespace "AboutMyEnvironment".
- Status Bar:** Shows "Projects: 2" and other standard status bar information.

The screenshot shows a GitHub repository page for 'markjprice / cs12dotnet8'. The repository is public and contains 1 branch and 0 tags. The main file listed is 'README.md'. On the right side, there is an 'About' section describing the repository as the source code for the Packt Publishing book 'C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals' by Mark J. Price. It also shows statistics like 13 stars, 3 watching, and 1 fork. Below the 'About' section is a 'Releases' section which is currently empty.

The screenshot shows the Microsoft Learn .NET documentation page for the 'dotnet build' command. The page includes a sidebar with links to '.NET tools and diagnostics', '.NET SDK', '.NET CLI', and 'dotnet command reference'. The main content area features a large heading 'dotnet build', a brief description, and a 'Feedback' button. The URL in the browser bar is 'https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-build'.





Posts containing 'securestring' - X

https://stackoverflow.com/search?q=securestring

stackoverflow About Products For Teams Log in Sign up

Home PUBLIC Questions Tags Users Companies COLLECTIVES Explore Collectives TEAMS Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.

Search Results

Results for securestring Search options not deleted

500 results Relevance Newest More ▾

185 votes When would I need a SecureString in .NET? And why can't I just say: `SecureString password = new SecureString("password");` instead of `SecureString pass = new...` .net security encryption Richard Morgan 7,591 asked Sep 26, 2008 at 18:43

187 votes How to convert SecureString to System.String? All reservations about unsecuring your `SecureString` by creating a `System.String` out of it aside, how can it be done? How can I conver... 166k views Indeed Build your career in Front End Development with JavaScript CSS React

Microsoft Azure What will you create? Start building apps today. Free until you say otherwise. Try Azure free! Report this ad

Code search results

github.com/search?q=email+repo%3Adotnet%2Fruntime+language%3AC%23&type=code&ref=advsearch

email repo:dotnet/runtime language:C#

Filter by

- Code 99
- Issues 0
- Pull requests 0
- Discussions 0
- Commits 0
- Packages 0
- Wikis 0
- Paths

99 files (140 ms) in dotnet/runtime

```
5 {
6     [AttributeUsage(AttributeTargets.Property | AttributeTargets.Field |
7         AttributeTargets.Parameter,
8             AllowMultiple = false)]
9     public sealed class EmailAddressAttribute : DataTypeAttribute
10    {
11        public EmailAddressAttribute()
12            : base(DataType.EmailAddress)
```

Show 1 more match

A screenshot of a web browser window titled "Email Validation in C#". The URL is <https://chat.openai.com/?model=gpt-4>. The sidebar on the left shows a list of previous chats, including "Email Validation in C#", "Customize Zoom Waiting Room", "Configuring Preferences for AI", "UK A-level grade inflation", and "LinkedIn recommendation for". The main area shows a prompt from "MP" asking "write a c# function to validate an email address". ChatGPT responds with a C# code snippet:

```
cssharp
using System.Text.RegularExpressions;

public class EmailValidator
{
    public static bool IsValidEmail(string email)
    {
        if (string.IsNullOrWhiteSpace(email))
            return false;

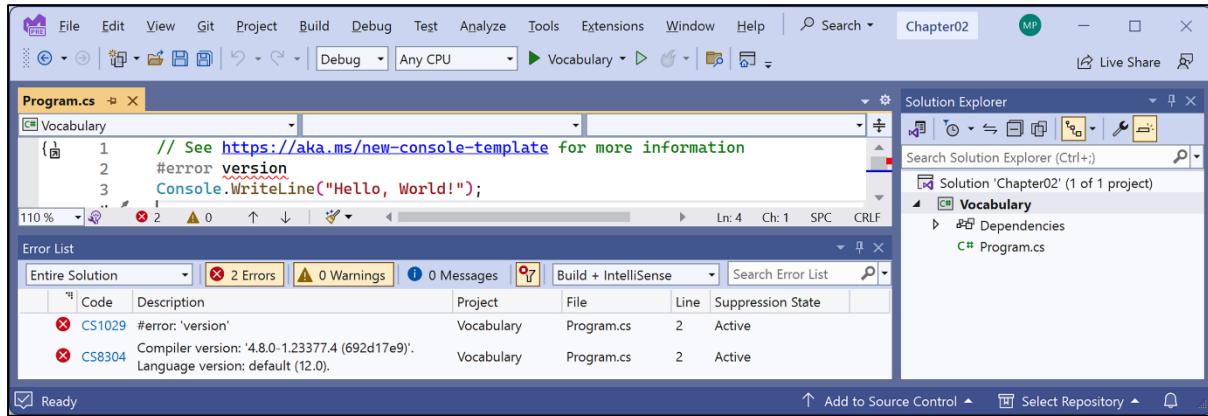
        // Regular expression for validating an Email
        string regex = "^[\\w+_\\.-]+@[\\w+_\\.-]+\\.[\\w]{2,}$";
        Regex pattern = new Regex(regex);
        return pattern.IsMatch(email);
    }
}
```

The message input field says "Send a message." and a note at the bottom says "ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT May 12 Version".

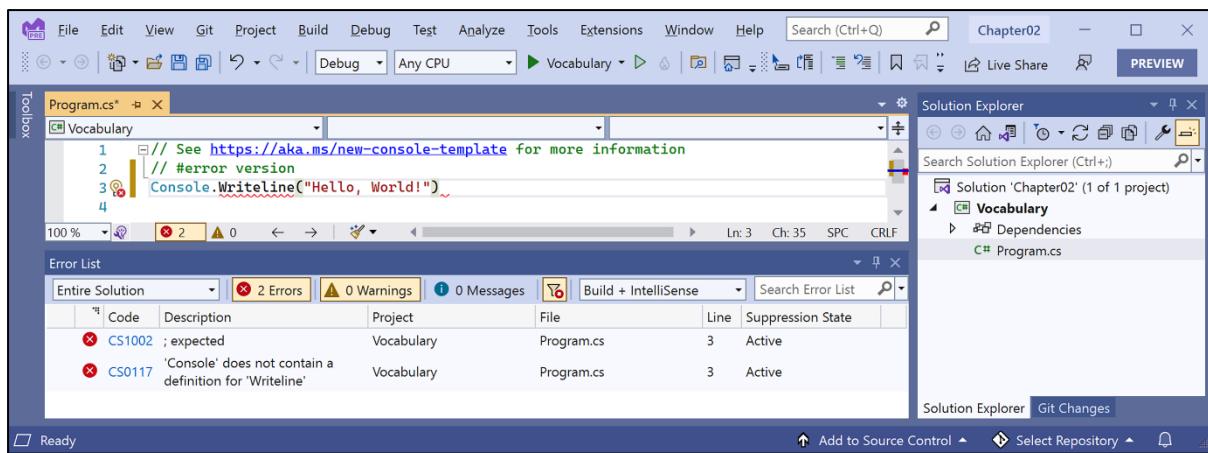
A screenshot of Microsoft Visual Studio. The title bar says "Chapter01". The main window shows the code editor with "Product.cs" open. The code defines a `HelloCS` namespace and a `Product` class with properties `Name`, `Price`, `Quantity`, and `Description`, and a constructor and `ToString` method. The Solution Explorer on the right shows a solution named "Chapter01" with projects "AboutMyEnvironment", "HelloCS" (which contains "Dependencies", "C# Product.cs", and "C# Program.cs"), and a file "HelloCS.cs".

A screenshot of a web browser showing a GitHub page for the "dotnet/installer" repository. The URL is <https://github.com/dotnet/installer#table>. The page displays a table of build versions. A context menu is open over the "dotnet-sdk-9.0.100-alpha.1.23424.7-win" file in the "Downloads" section, with options like "Delete", "Keep", "Report this file as safe", "Learn more", and "Copy download link".

Chapter 2: Speaking C#

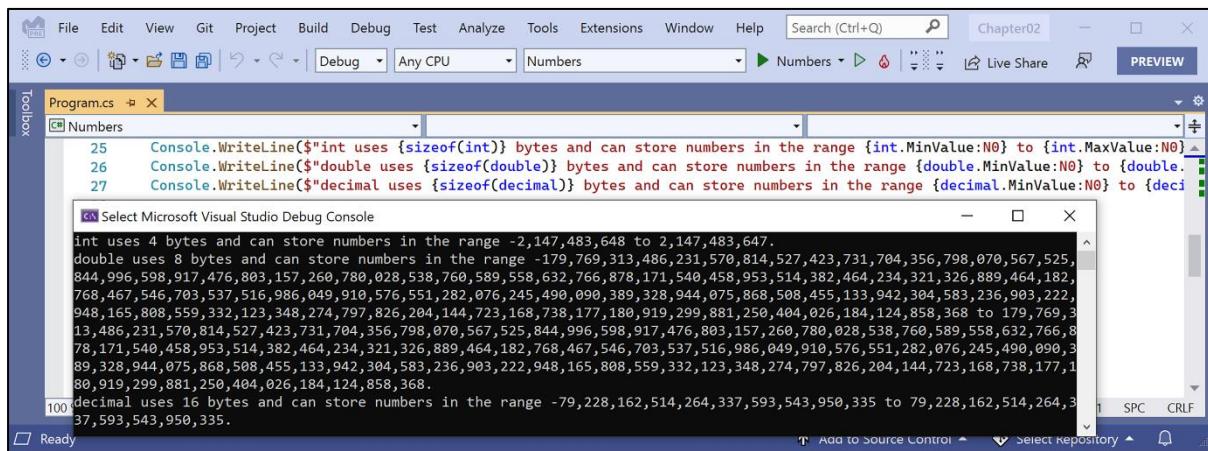


Visual Studio 2022	Visual Studio Code
<pre>// Loop through all the types in the assembly. foreach (TypeInfo t in a.DefinedTypes) { // Add up the counts of all the methods. methodCount += t.GetMethods().Length; }</pre>	<pre>// Loop through all the types in the assembly. foreach (TypeInfo t in a.DefinedTypes) { // Add up the counts of all the methods. methodCount += t.GetMethods().Length; }</pre>
<pre>// Loop through all the types in the assembly. foreach (TypeInfo t in a.DefinedTypes)...</pre>	<pre>// Loop through all the types in the assembly. foreach (TypeInfo t in a.DefinedTypes)...</pre>

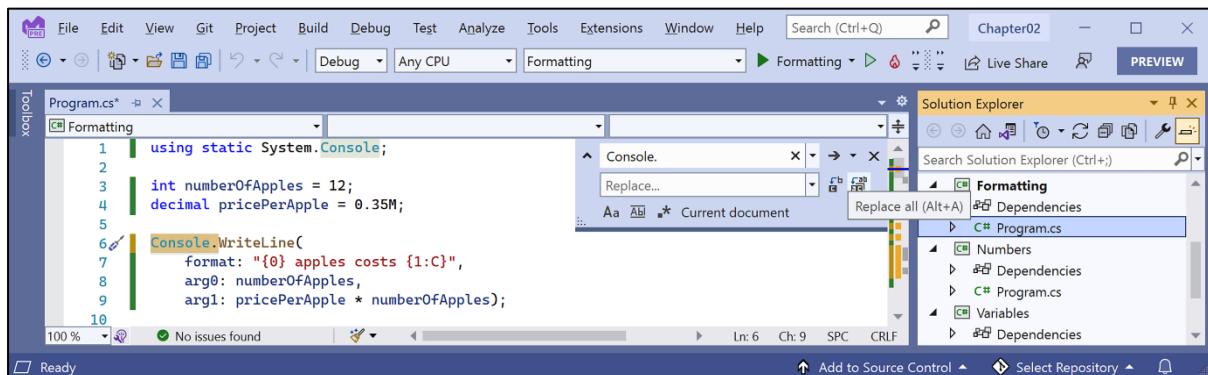
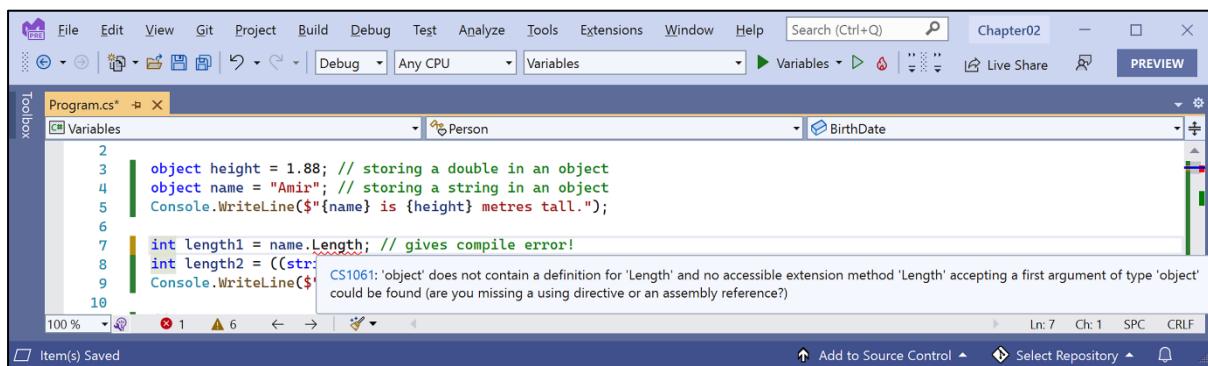


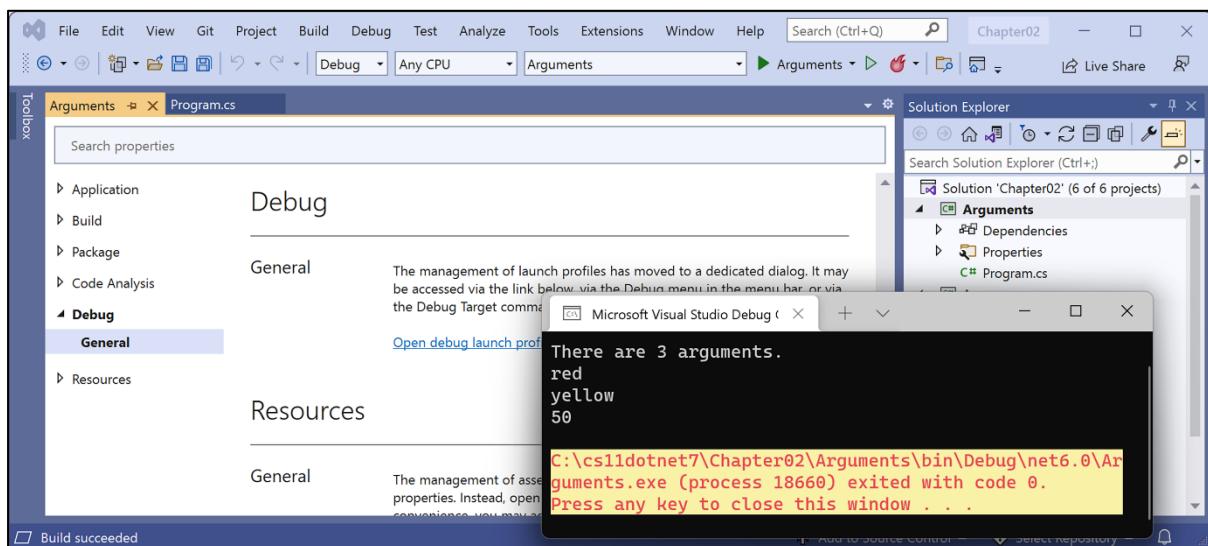
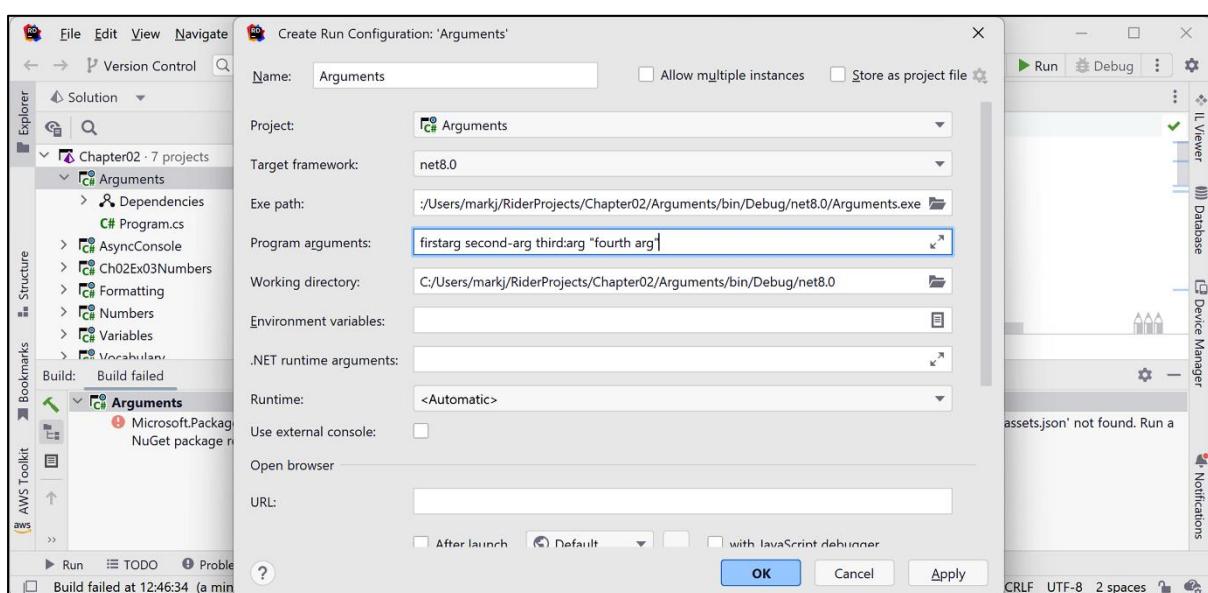
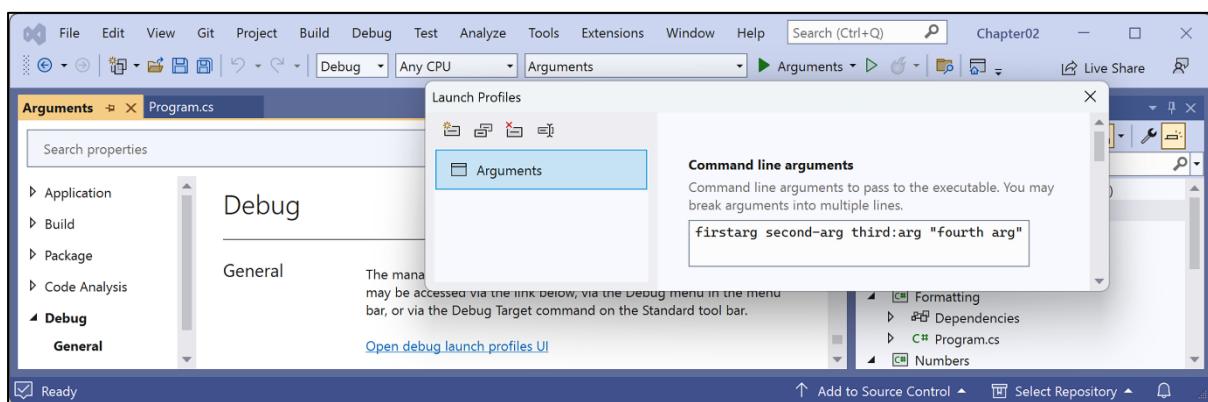
128	64	32	16	8	4	2	1
0	0	0	0	1	0	1	0

$$\begin{array}{ccccccccc|c} \hline 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 & \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \\ \hline \end{array} \cdot \begin{array}{cccc} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} \\ \hline 1 & 1 & 0 & 0 \end{array}$$



4	2	1	.	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$
0	0	0	.	0	0	0	1	1	0	0	1	1	0	0





A screenshot of Visual Studio Code showing a C# file named Program.cs. The code reads command-line arguments and sets the background color and cursor size. The terminal shows the output of running the program with arguments 'red' and 'yellow'. An unhandled exception is thrown because the operation is not supported on the current platform.

```
BackgroundColor = (ConsoleColor)Enum.Parse(
    enumType: typeof(ConsoleColor),
    value: args[1],
    ignoreCase: true);

CursorSize = int.Parse(args[2]);

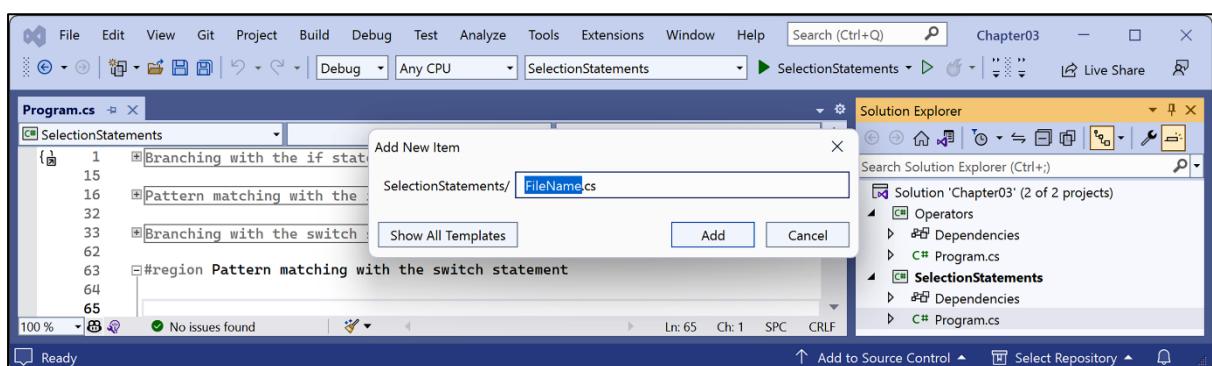
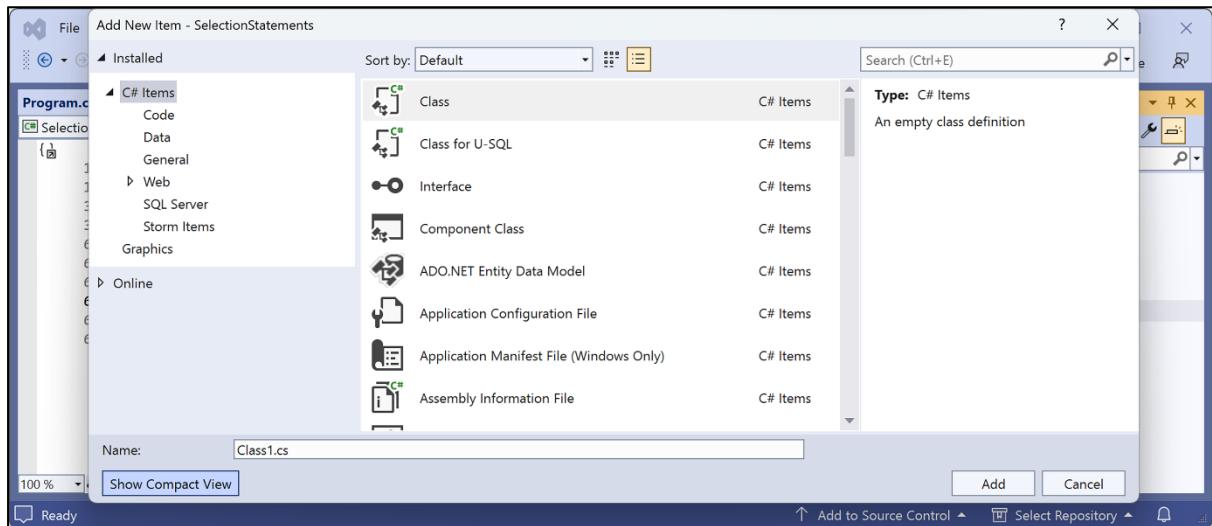
Arguments > Program.cs > {} Arguments > Arguments.Program > Main(string[] args)
29
30
31
32
33
34

TERMINAL PROBLEMS 6 OUTPUT DEBUG CONSOLE 3: bash
Marks-MacBook-Pro-13:Arguments markjprice$ dotnet run red yellow 50
There are 3 arguments.
red
yellow
50
Unhandled exception. System.PlatformNotSupportedException: Operation is not supported on this platform.
at System.ConsolePal.set_CursorSize(Int32 value)
at System.Console.set_CursorSize(Int32 value)
at Arguments.Program.Main(String[] args) in /Users/markjprice/Code/Chapter02/Arguments/Program.cs:line 36
Marks-MacBook-Pro-13:Arguments markjprice$
```

A screenshot of the Microsoft Visual Studio Debug Console showing a table of memory types and their byte sizes, minimum, and maximum values.

Type	Byte(s) of memory	Min	Max
sbyte	1	-128	127
byte	1	0	255
short	2	-32768	32767
ushort	2	0	65535
int	4	-2147483648	2147483647
uint	4	0	4294967295
long	8	-9223372036854775808	9223372036854775807
ulong	8	0	18446744073709551615
Int128	16	-170141183460469231731687303715884105728	170141183460469231731687303715884105727
UInt128	16	0	340282366920938463463374607431768211455
Half	2	-65500	65500
float	4	-3.4028235E+38	3.4028235E+38
double	8	-1.7976931348623157E+308	1.7976931348623157E+308
decimal	16	-79228162514264337593543950335	79228162514264337593543950335

Chapter 3: Controlling Flow, Converting Types, and Handling Exceptions



0	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>Alpha</td><td>Beta</td><td>Gamma</td></tr></table>	0	1	2	Alpha	Beta	Gamma		
0	1	2							
Alpha	Beta	Gamma							
1	<table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Anne</td><td>Ben</td><td>Charlie</td><td>Doug</td></tr></table>	0	1	2	3	Anne	Ben	Charlie	Doug
0	1	2	3						
Anne	Ben	Charlie	Doug						
2	<table border="1"><tr><td>0</td><td>1</td></tr><tr><td>Aardvark</td><td>Bear</td></tr></table>	0	1	Aardvark	Bear				
0	1								
Aardvark	Bear								

The screenshot shows the Microsoft Visual Studio IDE. The code editor displays the following C# code:

```
60 //#region Disabling compiler overflow checks with the unchecked statement
61
62 int y_ = int.MaxValue + 1;
63
64 //#endregion
```

A tooltip is displayed over the line `int y_ = int.MaxValue + 1;`, containing the following information:

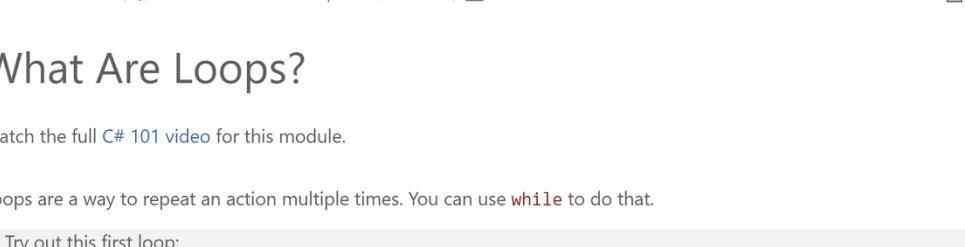
- (constant) const int int.MaxValue = 2147483647
- Represents the largest possible value of an int. This field is constant.

The tooltip also includes the error message: CS2020: The operation overflows at compile time in checked mode.

The Solution Explorer on the right shows the project structure:

- Solution 'Chapter03' (6 of 6 projects)
 - Arrays
 - CastingConverting
 - HandlingExceptions (selected)
 - Dependencies

```
Microsoft Visual Studio Debug Console + ^ - □ ×  
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz,  
11, Fizz, 13, 14, FizzBuzz, 16, 17, Fizz, 19, Buzz,  
Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29, FizzBuzz,  
31, 32, Fizz, 34, Buzz, Fizz, 37, 38, Fizz, Buzz,  
41, Fizz, 43, 44, FizzBuzz, 46, 47, Fizz, 49, Buzz,  
Fizz, 52, 53, Fizz, Buzz, 56, Fizz, 58, 59, FizzBuzz,  
61, 62, Fizz, 64, Buzz, Fizz, 67, 68, Fizz, Buzz,  
71, Fizz, 73, 74, FizzBuzz, 76, 77, Fizz, 79, Buzz,  
Fizz, 82, 83, Fizz, Buzz, 86, Fizz, 88, 89, FizzBuzz,  
91, 92, Fizz, 94, Buzz, Fizz, 97, 98, Fizz, Buzz
```



File Edit Selection View Go Run Terminal Help • Untitled-1.ipynb - Visual Studio Code

Untitled-1.ipynb

+ Code + Markdown | ▶ Run All ⌘ Clear All Outputs ⌘ Restart | Variables Outline ...

.NET Interactive

What Are Loops?

Watch the full [C# 101](#) video for this module.

Loops are a way to repeat an action multiple times. You can use `while` to do that.

Try out this first loop:

```
int counter = 0;
while (counter < 10)
{
    Console.WriteLine($"Hello World! The counter is {counter}");
    counter++;
}
```

csharp - C# Script Code

Chapter 4: Writing, Debugging, and Testing Functions

```
using static System.Console;
WriteLine(* Top-level function
WhatsMyNamespace(); // Call
```

```
partial class Program
{
    static void WhatsMyNamespace()
    {
        WriteLine("Namespace of Program");
        arg0: typeof(Program).Name
    }
}
```

Solution Explorer:

- Solution 'Chapter04' (1 of 1 project)
 - TopLevelFunctions
 - Dependencies
 - Program.cs
 - C# Program.Functions.cs
 - Program
 - <top-level-statements-entry-point> : void
 - WhatsMyNamespace() : void

```
static void RunCardinalToOrdinal()
{
    for (int number = 1; number <= 40; number++)
    {
        Write($"{CardinalToOrdinal(number)} ");
    }
    WriteLine();
}
```

string Program.CardinalToOrdinal(int number)
Pass a 32-bit integer and it will be converted into its ordinal equivalent.
number: Number is a cardinal value e.g. 1, 2, 3, and so on.

Solution Explorer:

- Solution 'Chapter04' (1 of 1 project)
 - WritingFunctions
 - Dependencies
 - Program.cs
 - C# Program.Functions.cs

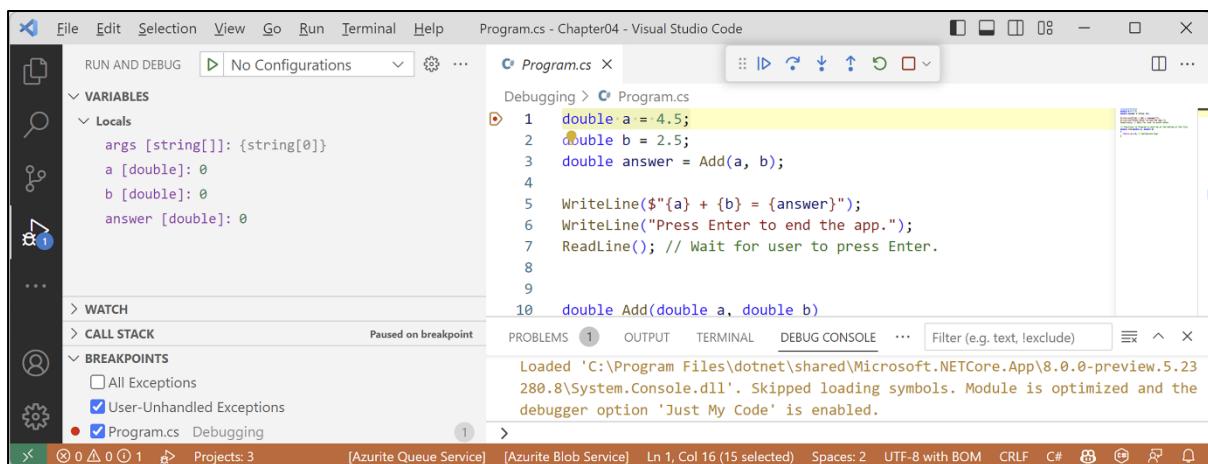
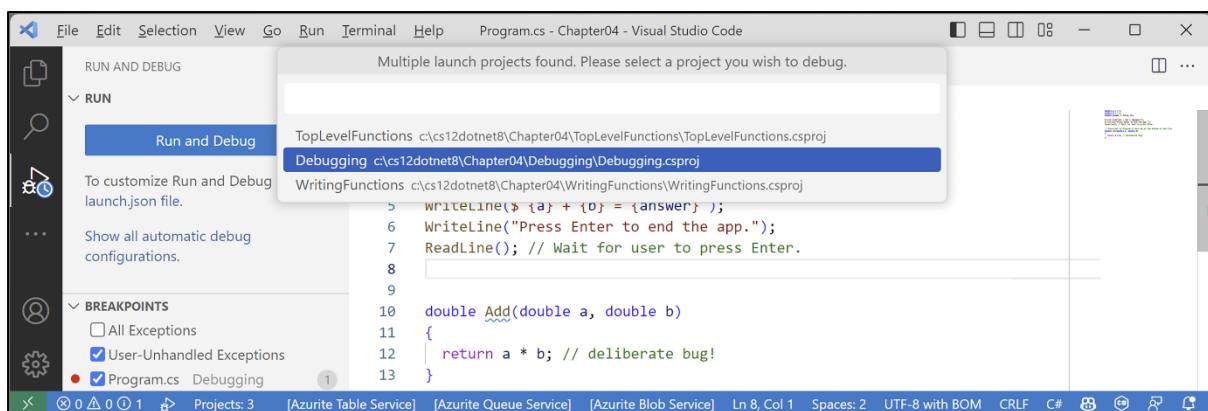
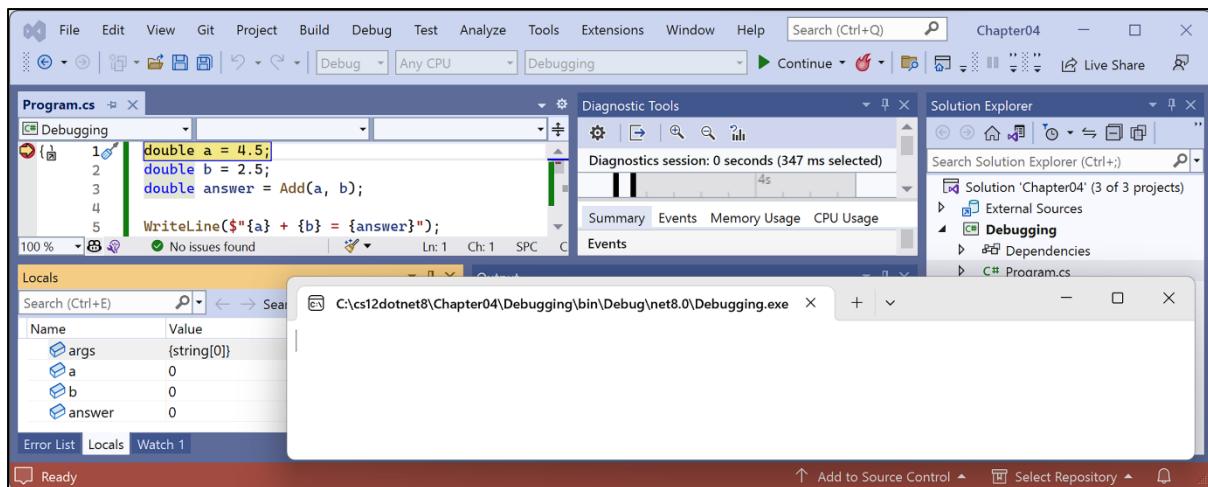
```
double a = 4.5;
double b = 2.5;
double answer = Add(a, b);

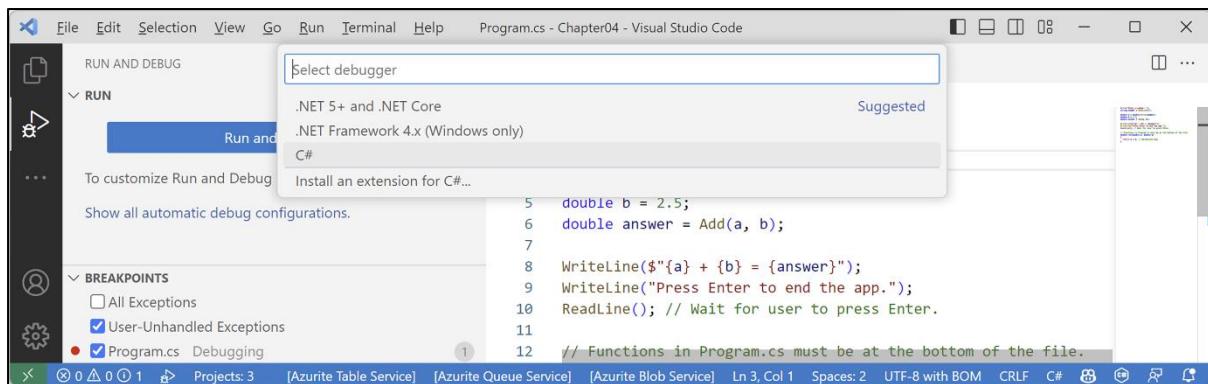
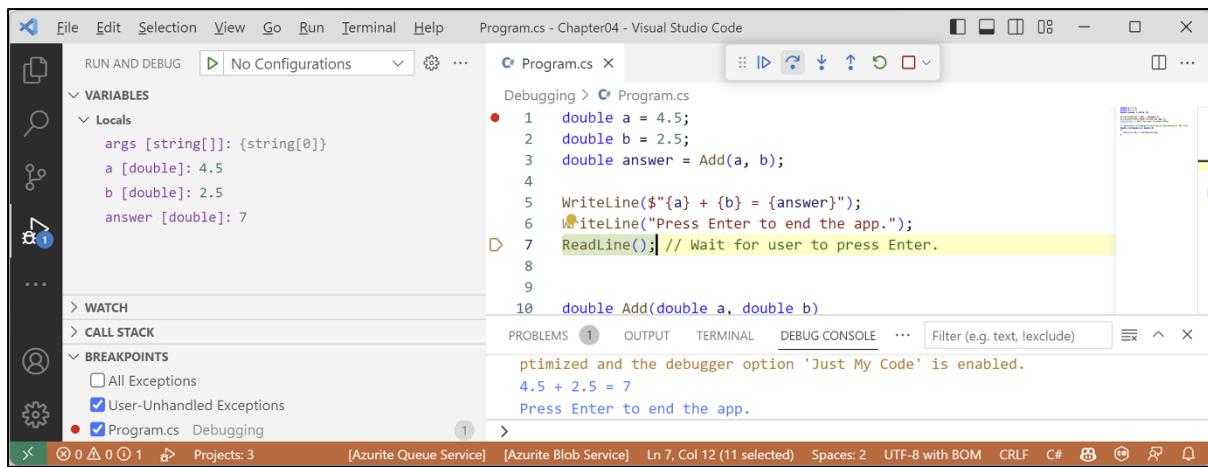
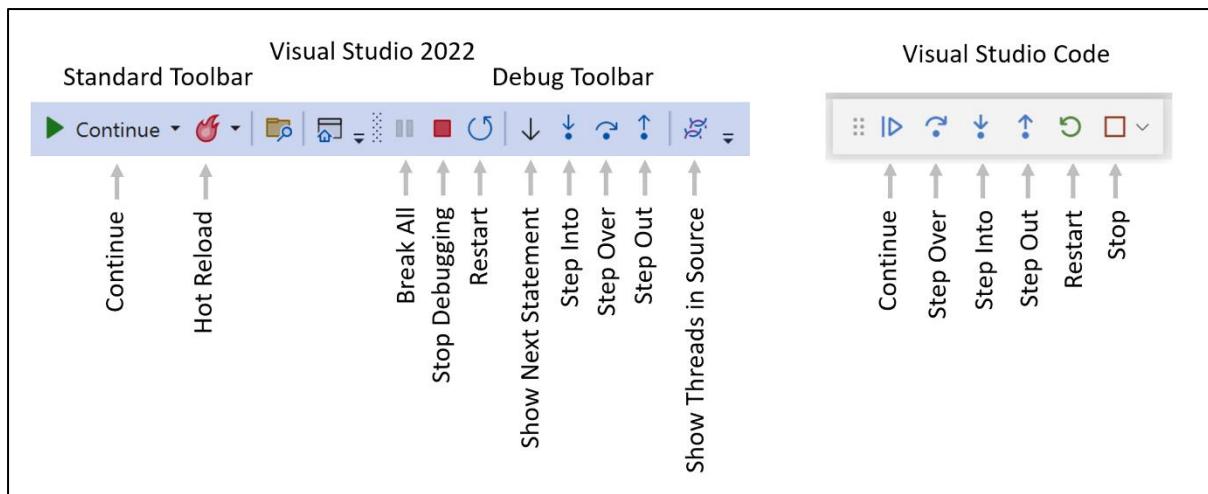
WriteLine($"{a} + {b} = {answer}");
WriteLine("Press Enter to end the app.");
ReadLine(); // Wait for user to press Enter.

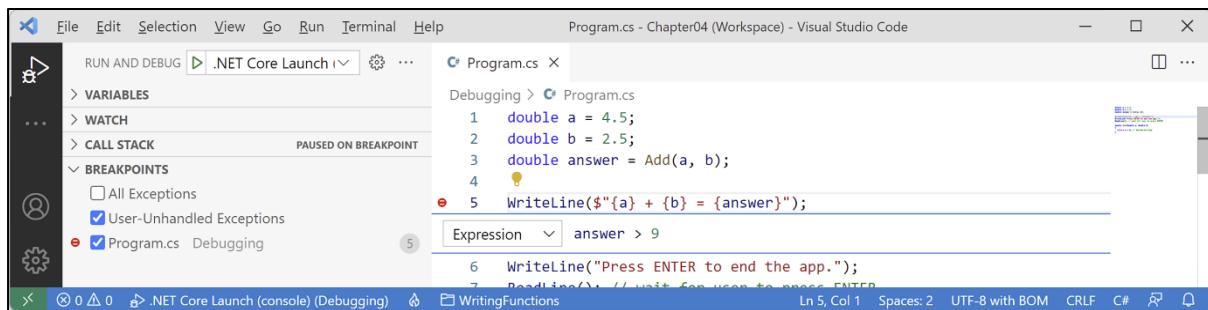
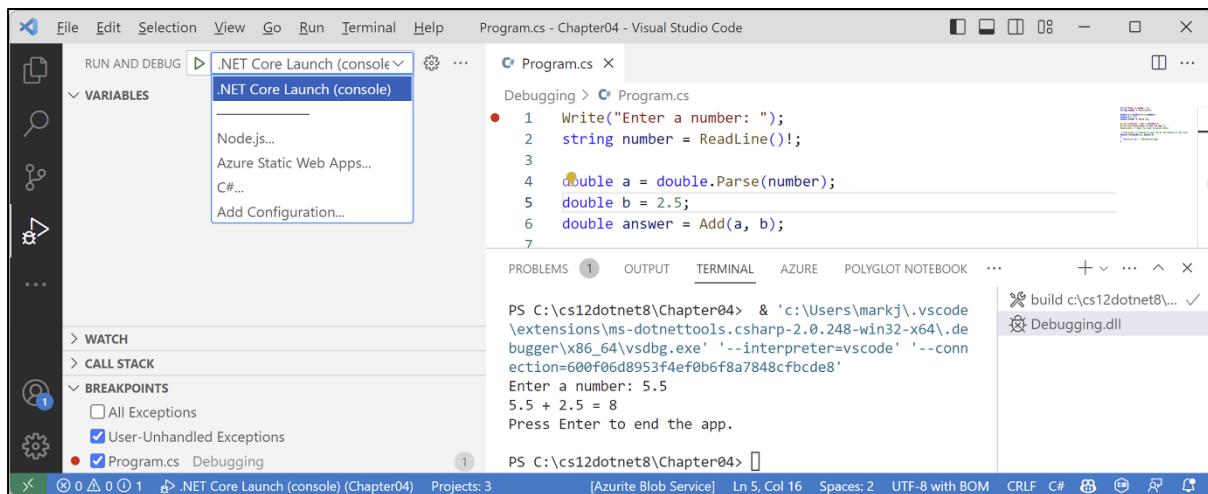
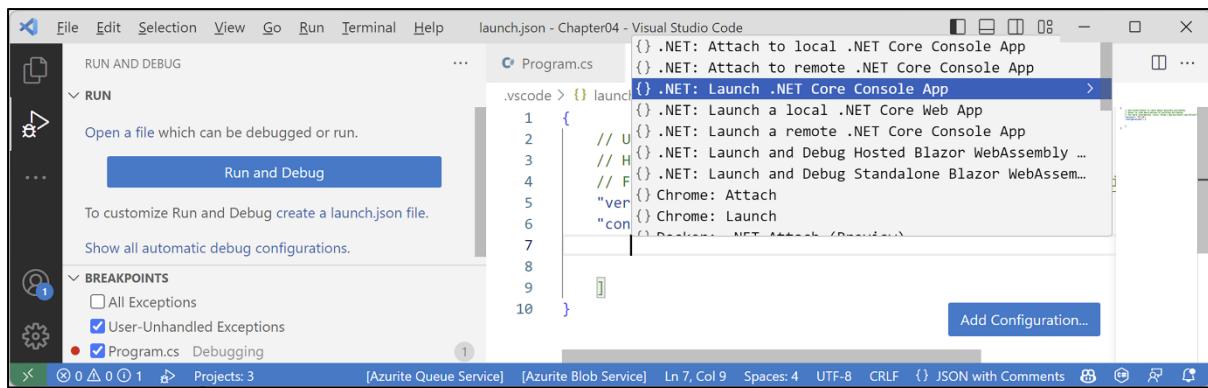
double Add(double a, double b)
{
    return a * b; // deliberate bug!
}
```

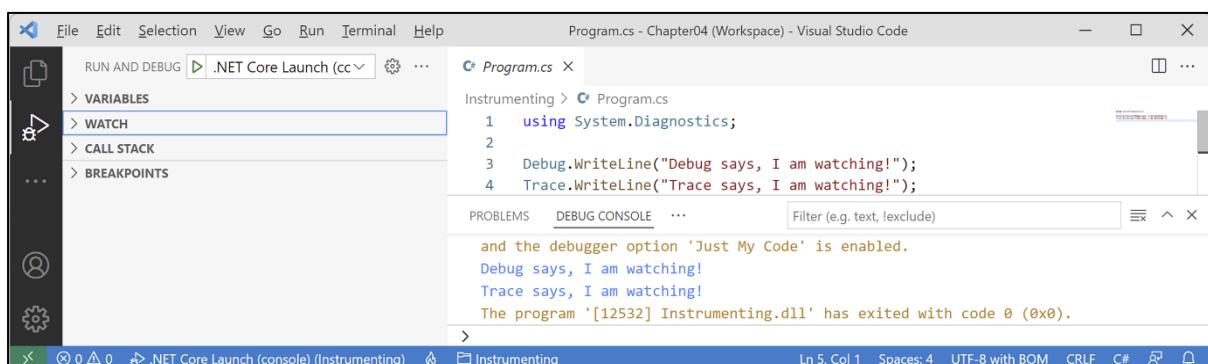
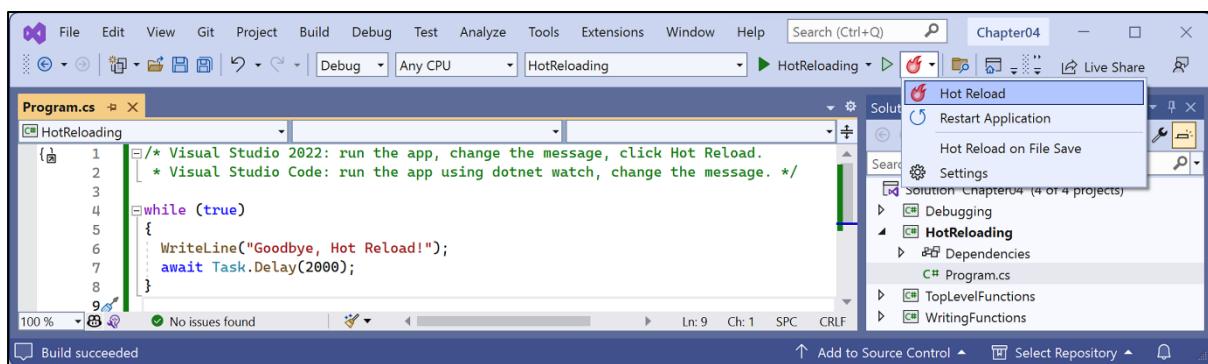
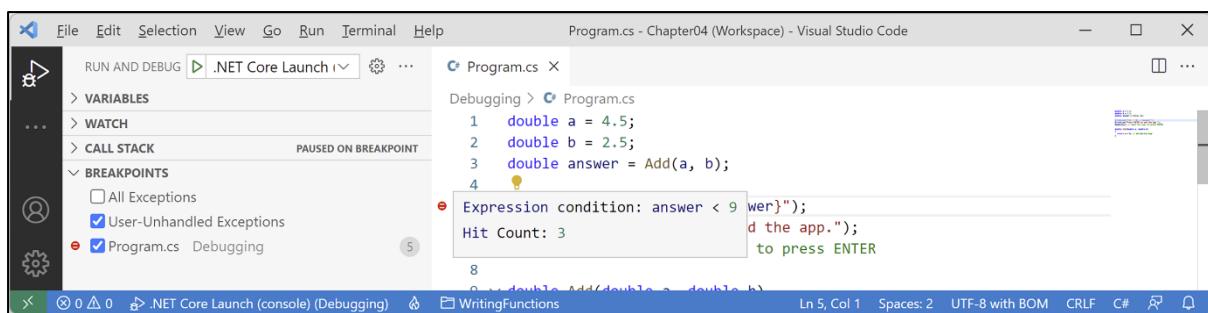
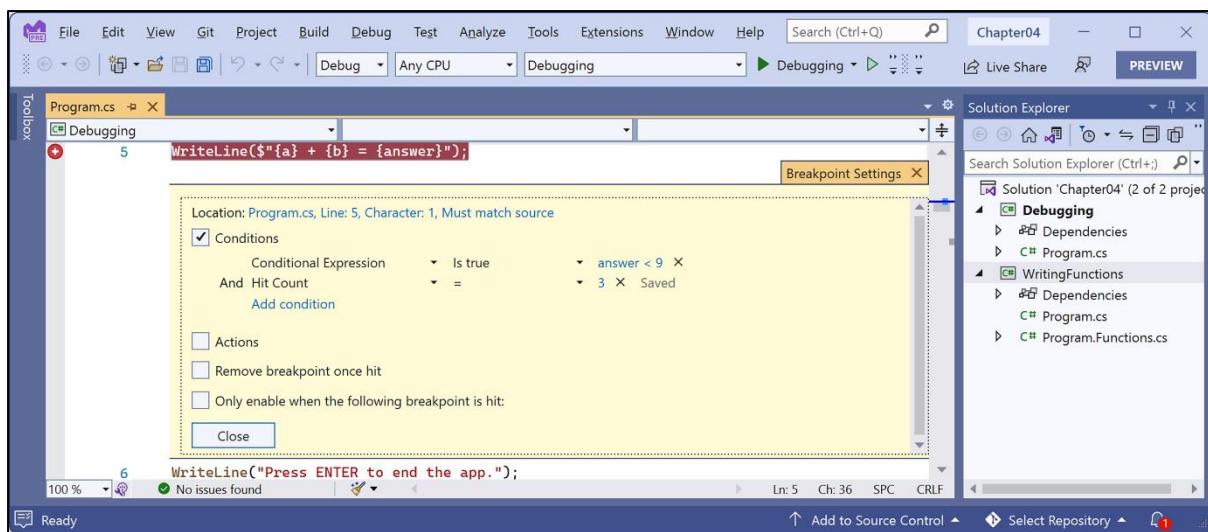
Solution Explorer:

- Solution 'Chapter04' (3 of 3 projects)
 - Debugging
 - Dependencies
 - Program.cs
 - TopLevelFunctions
 - WritingFunctions









The screenshot shows the Microsoft Visual Studio IDE interface. In the top navigation bar, the 'File', 'Edit', 'View', 'Git', 'Project', 'Build', 'Debug', 'Test', 'Analyze', 'Tools', 'Extensions', and 'Window' menus are visible. A search bar at the top right contains the text 'Search (Ctrl+Q)'. To the right of the search bar are buttons for 'Chapter04' and 'PREVIEW'. On the left side, there's a 'Toolbox' tab and a 'Toolbox' icon. The main workspace shows the 'Instrumenting' project in the Solution Explorer. The 'Program.cs' file is open, displaying the following code:

```
using System.Diagnostics;  
Debug.WriteLine("Debug says, I am watching!");  
Trace.WriteLine("Trace says, I am watching!");
```

The status bar at the bottom indicates '100 %' completion, 'No issues found', and file statistics: Ln: 38 Ch: 3 SPC CRLF.

In the 'Output' window, the message 'Show output from: Debug' is selected. The output pane displays the following text:

```
'Instrumenting.exe' (CoreCLR: DefaultDomain): Loaded 'C:\Program  
'Instrumenting.exe' (CoreCLR: clrhost): Loaded 'C:\Code\Chapter04  
'Instrumenting.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
'Instrumenting.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
'Instrumenting.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
Debug says, I am watching!  
Trace says, I am watching!
```

A separate 'Microsoft Visual Studio Debug Console' window is open, showing the application's output:

```
C:\Code\Chapter04\Instrumenting\bin\Debug\net6.0\Instrumenting.exe (process 16640) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

The screenshot shows the Microsoft Visual Studio interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). Below the menu is a toolbar with various icons. The solution configuration dropdown shows 'Release' and 'Any CPU'. The 'Instrumenting' dropdown is also visible. The main code editor window displays the following C# code:

```
10 Trace.Listeners.Add(logFile);
11
12 #if DEBUG
13 // Text writer is buffered, so this option calls
14 // Flush() on all listeners after writing.
15 Trace.AutoFlush = true;
16 #endif
17
18 Debug.WriteLine("Debug says, I am watching!");
19 Trace.WriteLine("Trace says, I am watching!");
```

The status bar at the bottom indicates 100% completion, no issues found, Ln: 20, Ch: 1, SPC, and CRLF.

The Solution Explorer pane on the right lists the projects in the solution: 'Chapter04' (5 of 5 projects), 'Debugging', 'HotReloading', 'Instrumenting' (selected), 'Dependencies', 'Program.cs' (selected), 'TopLevelFunctions', and 'WritingFunctions'.

The screenshot shows the Visual Studio interface with the title bar "Chapter04". The main window displays the "NuGet: Instrumenting" tab. In the center, there's a search bar with "Microsoft.Extensions.Configuration.Binder" typed in, and a checkbox labeled "Include prerelease" is checked. To the right, the results for "Microsoft.Extensions.Configuration.Binder" are shown, listing two packages: "Microsoft.Extensions.Configuration.Binder" (version 8.0.0-preview.5.23280.8) and "RockLib.Configuration.ObjectFactory" (version 2.0.2). Both packages have their descriptions and download counts displayed. On the far right, a detailed view of the first package is open, showing its version as "Latest prerelease 8.0.0-", an "Install" button, and an "Options" link. At the bottom left, a note about package licensing is present, and at the bottom right, there are links for "Add to Source Control" and "Select Repository".

The screenshot shows the Visual Studio IDE during a debug session. The title bar says "Chapter04". The main window displays a C# file named "Program.cs" with the following code:

```
Instrumenting
configuration.GetSection("PacktSwitch").Bind(ts);
```

The Locals window is open, showing the state of the variable "ts" and its properties:

Name	Type	Value
ts	System.Diagnostics.TraceSwitch	(System.Diagnostics.TraceSwitch)
Attributes	System.Collections.Specialized.StringDictionary	(System.Collections.Specialized.StringDictionary)
DefaultValue	System.Collections.Specialized.StringDictionary	"0"
Description	System.Collections.Specialized.StringDictionary	"This switch is set via a JSON config."
DisplayName	System.Collections.Specialized.StringDictionary	"PacktSwitch"
Level	System.Diagnostics.TraceLevel	Off
SwitchSetting	System.Diagnostics.TraceLevel	0
TraceError	System.Diagnostics.TraceLevel	false
TraceInfo	System.Diagnostics.TraceLevel	false
TraceVerbose	System.Diagnostics.TraceLevel	false
TraceWarning	System.Diagnostics.TraceLevel	false
Value	System.Collections.Specialized.StringDictionary	"0"

The Solution Explorer on the right shows a solution named "Chapter04" containing five projects: External Sources, Debugging, HotReloading, Instrumenting (selected), Dependencies, appsettings.json, C# Program.cs, TopLevelFunctions, and WritingFunctions.

This screenshot shows the same Visual Studio session after some modifications. The code now includes two additional lines:

```
Instrumenting
Console.WriteLine($"Trace switch value: {ts.Value}");
Console.WriteLine($"Trace switch level: {ts.Level}");
```

The Locals window shows the updated values for "ts" and its properties:

Name	Type	Value
ts	System.Diagnostics.TraceSwitch	(System.Diagnostics.TraceSwitch)
Attributes	System.Collections.Specialized.StringDictionary	(System.Collections.Specialized.StringDictionary)
DefaultValue	System.Collections.Specialized.StringDictionary	"0"
Description	System.Collections.Specialized.StringDictionary	"This switch is set via a JSON config."
DisplayName	System.Collections.Specialized.StringDictionary	"PacktSwitch"
Level	System.Diagnostics.TraceLevel	Info
SwitchSetting	System.Diagnostics.TraceLevel	3
TraceError	System.Diagnostics.TraceLevel	true
TraceInfo	System.Diagnostics.TraceLevel	true
TraceVerbose	System.Diagnostics.TraceLevel	false
TraceWarning	System.Diagnostics.TraceLevel	true
Value	System.Collections.Specialized.StringDictionary	"Info"

This screenshot shows Visual Studio with a unit test project open. The title bar says "Chapter04" and "PREVIEW". The main window displays the "Test Explorer" window, which lists the following test results:

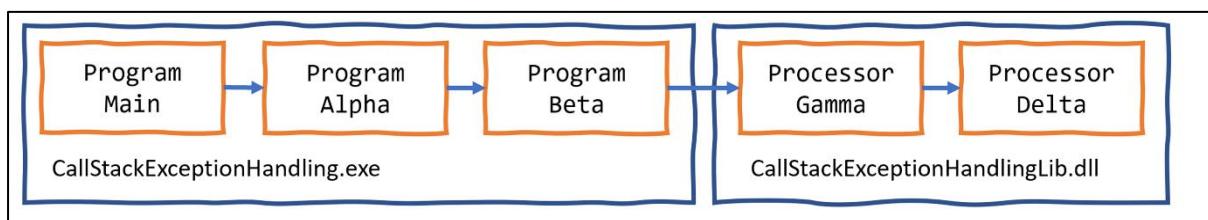
Test	Duration	Traits	Error Message
CalculatorLibUnitTests (2)	8 ms		
CalculatorLibUnitTests (2)	8 ms		
CalculatorLibUnitTests (2)	8 ms		
TestAdding2And2	< 1 ms		
TestAdding2And3	8 ms		Assert.Equal() Failure Expected: 5 Actual: 6

The "Diagnostic Tools" window on the right provides a summary of the test results:

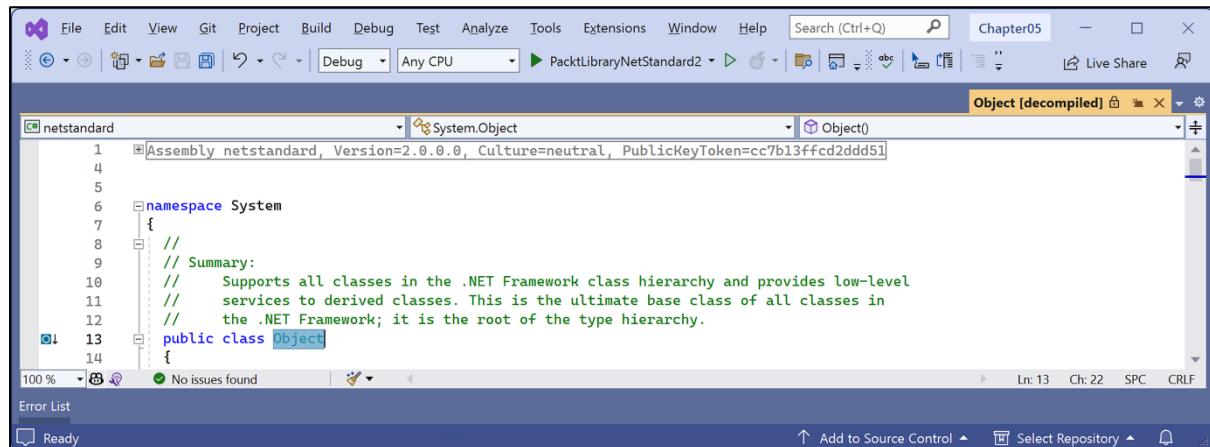
- Group Summary: CalculatorLibUnitTests, Tests in group: 2, Total Duration: 8 ms
- Outcomes: 1 Passed, 1 Failed

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** CalculatorUnitTests.cs - Chapter04 - Visual Studio Code
- Left Sidebar:** TESTING, Filter (e.g. text, !exclude, @tag), 1/2 tests passed (50.0%).
- Code Editor:** CalculatorUnitTests.cs (Line 33: Assert.Equal(expected, actual);) with a red error underline.
- Output Panel:** PROBLEMS tab (1), The test run did not record any output.
- Terminal:** Test run at 7/1/2023, 10:36:20 AM.
- Bottom Status Bar:** Projects: 7, [Azurite Table Service], [Azurite Queue Service], [Azurite Blob Service], Ln 33, Col 1, Spaces: 4, UTF-8, CRLF, C#, etc.

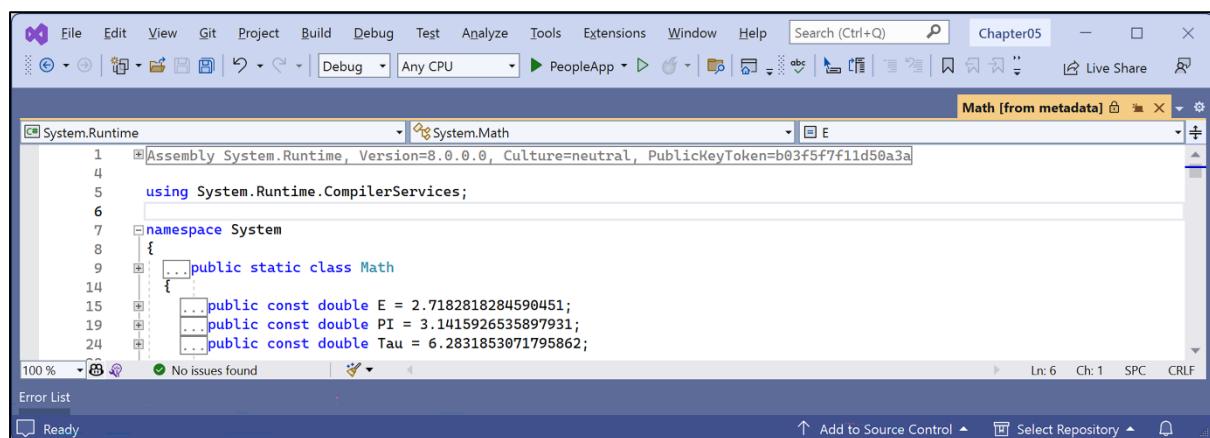


Chapter 5: Building Your Own Types with Object-Oriented Programming



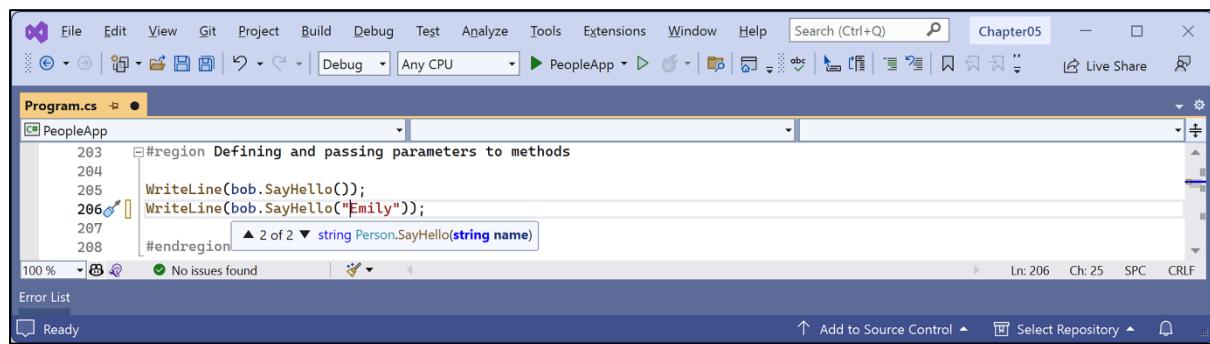
The screenshot shows the Visual Studio IDE with the title bar "Chapter05". The main window displays the decompiled code for the `Object` class from the `System` namespace. The assembly is `netstandard` (Version 2.0.0.0). The code includes a summary block stating: "Supports all classes in the .NET Framework class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all classes in the .NET Framework; it is the root of the type hierarchy." The code block starts with `public class Object`.

```
1 // Assembly netstandard, Version=2.0.0.0, Culture=neutral, PublicKeyToken=cc7b13ffcd2ddd51
2
3
4
5
6     namespace System
7     {
8         // ...
9         // Summary:
10        //     Supports all classes in the .NET Framework class hierarchy and provides low-level
11        //     services to derived classes. This is the ultimate base class of all classes in
12        //     the .NET Framework; it is the root of the type hierarchy.
13         public class Object
14     }
```



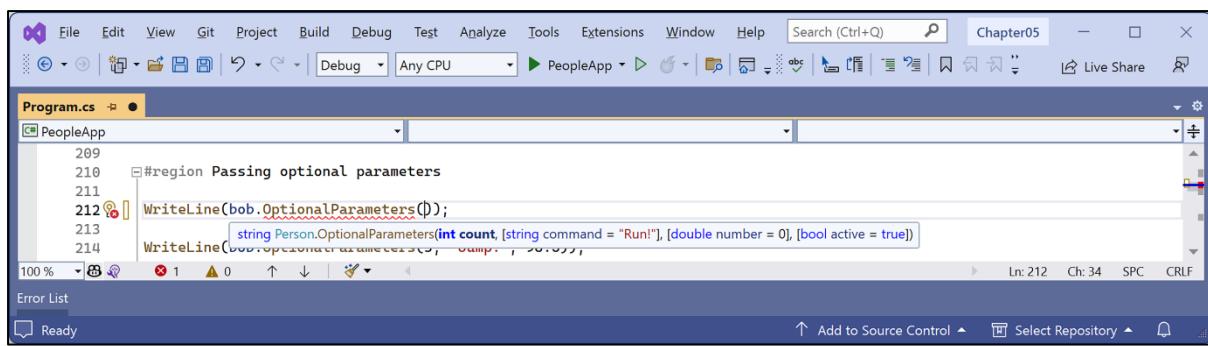
The screenshot shows the Visual Studio IDE with the title bar "Chapter05". The main window displays the decompiled code for the `Math` class from the `System` namespace. The assembly is `System.Runtime` (Version 8.0.0.0). The code includes constants for Euler's number (E), Pi, and Tau. The code block starts with `public static class Math`.

```
1 // Assembly System.Runtime, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
2
3
4     using System.Runtime.CompilerServices;
5
6     namespace System
7     {
8         public static class Math
9         {
10             public const double E = 2.7182818284590451;
11             public const double PI = 3.1415926535897931;
12             public const double Tau = 6.2831853071795862;
13         }
14     }
```



The screenshot shows the Visual Studio IDE with the title bar "Chapter05". The main window displays the code for `Program.cs` in the `PeopleApp` project. The code includes a region titled "Defining and passing parameters to methods". It contains two `WriteLine` statements: one calling `bob.SayHello()` and another calling `bob.SayHello("Emily")`. The code block starts with `#region Defining and passing parameters to methods`.

```
203 #region Defining and passing parameters to methods
204
205     WriteLine(bob.SayHello());
206     WriteLine(bob.SayHello("Emily"));
207
208 #endregion
```



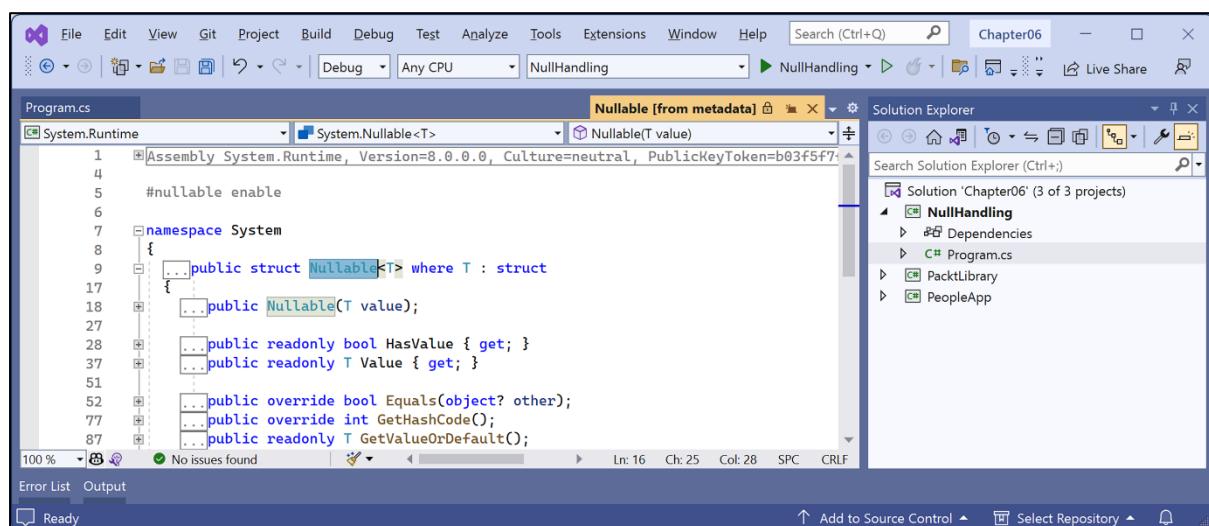
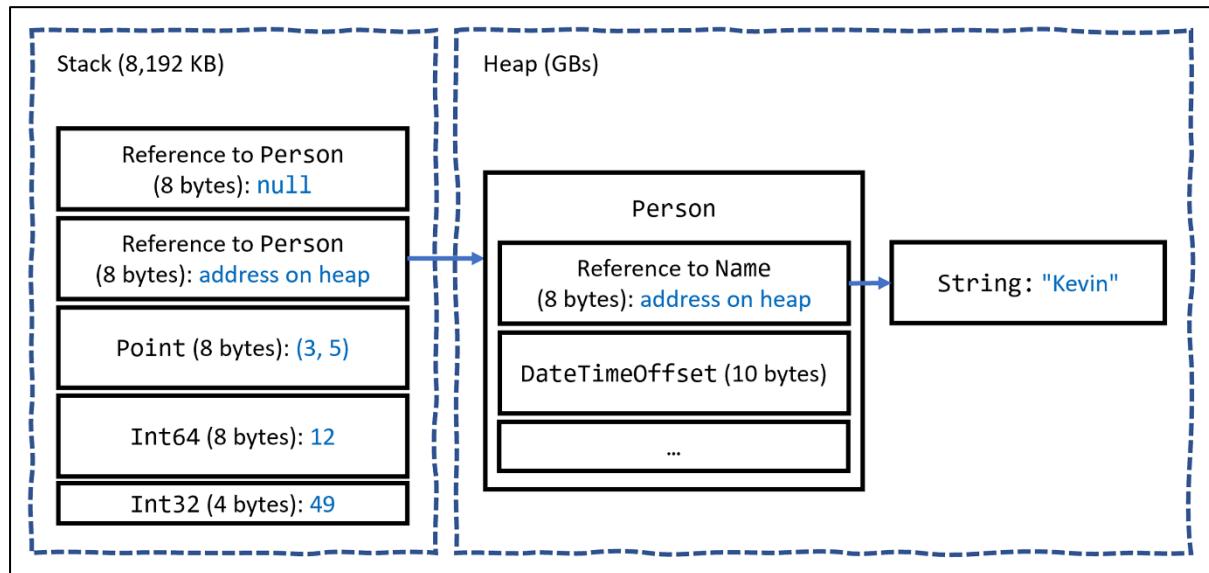
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Chapter05

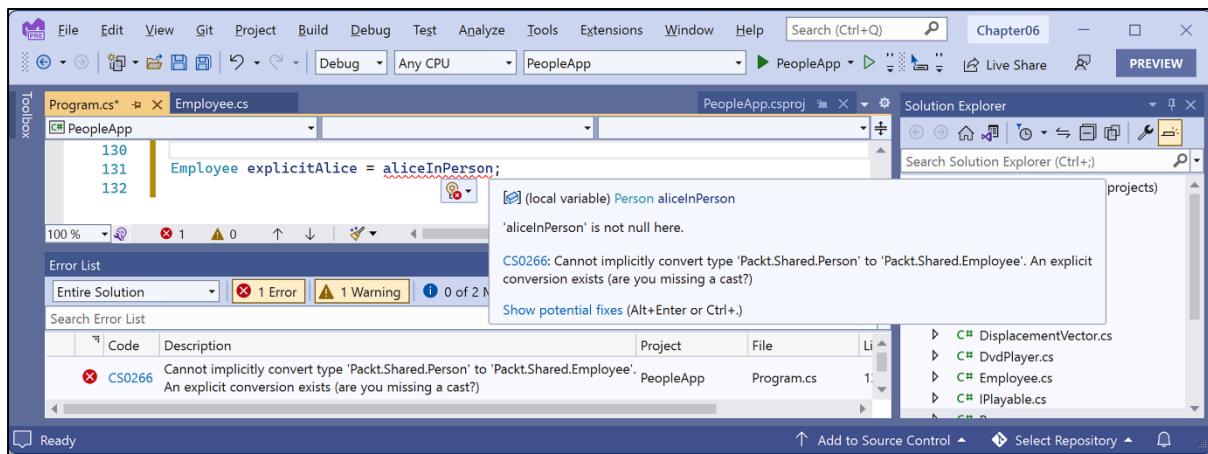
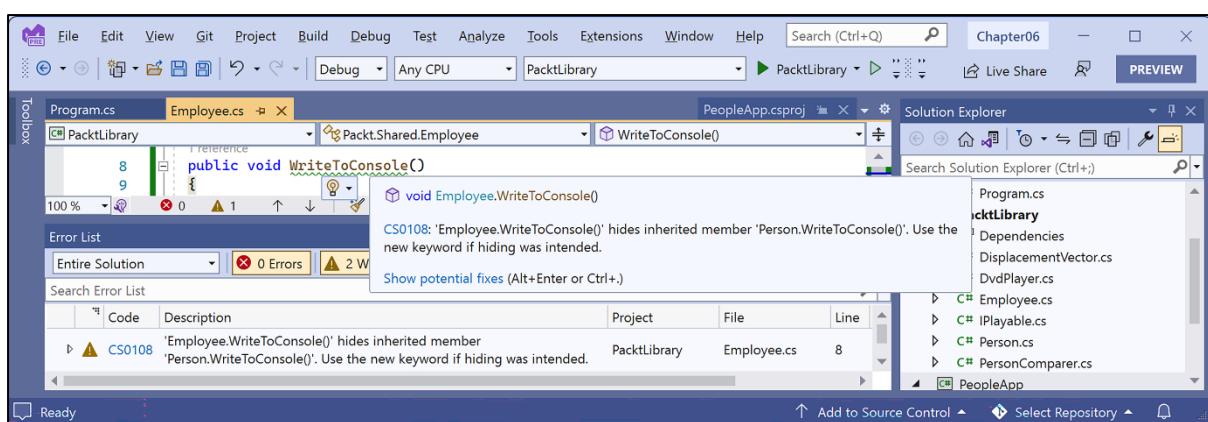
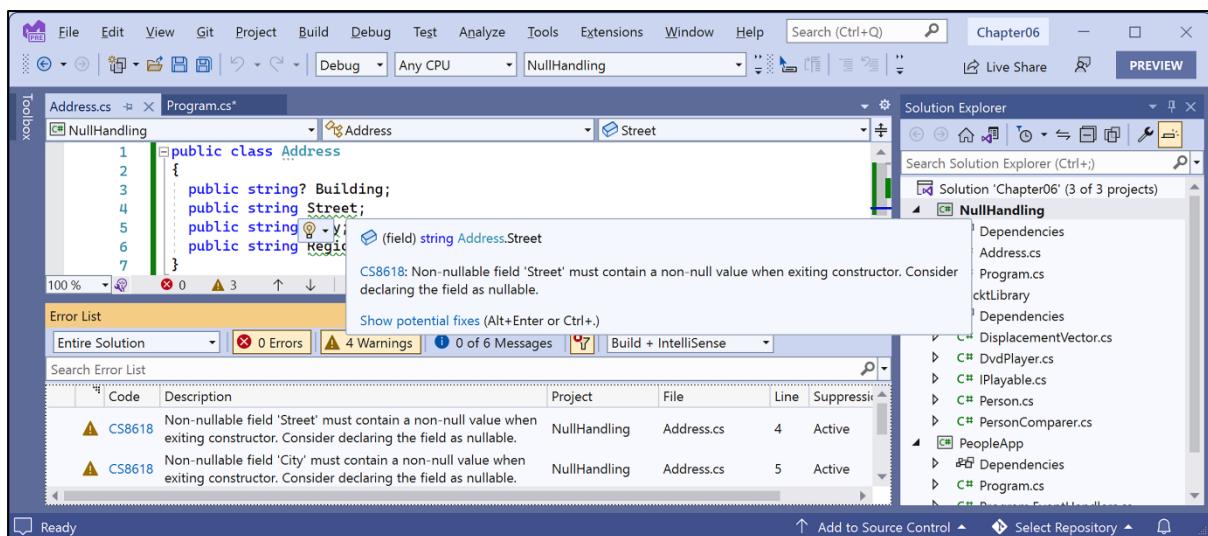
Program.cs

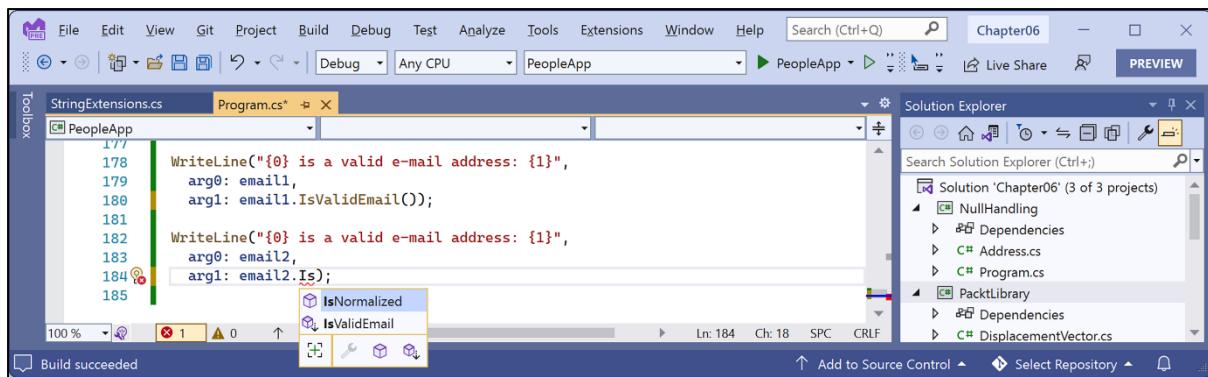
```
209
210  #region Passing optional parameters
211
212  WriteLine(bob.OptionalParameters());
213  // string Person.OptionalParameters(int count, string command = "Run!", double number = 0, bool active = true)
214  WriteLine(bob.OptionalParameters("Run!", 0, true));
```

100 % Error List Ready Add to Source Control Select Repository

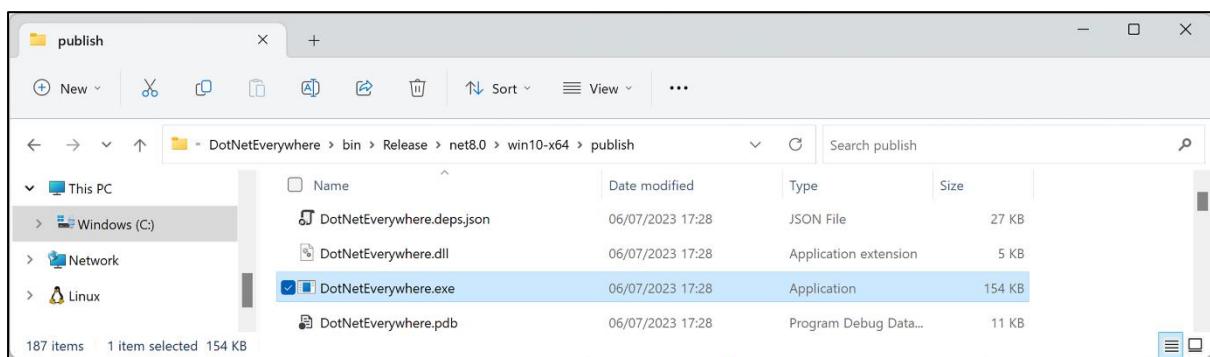
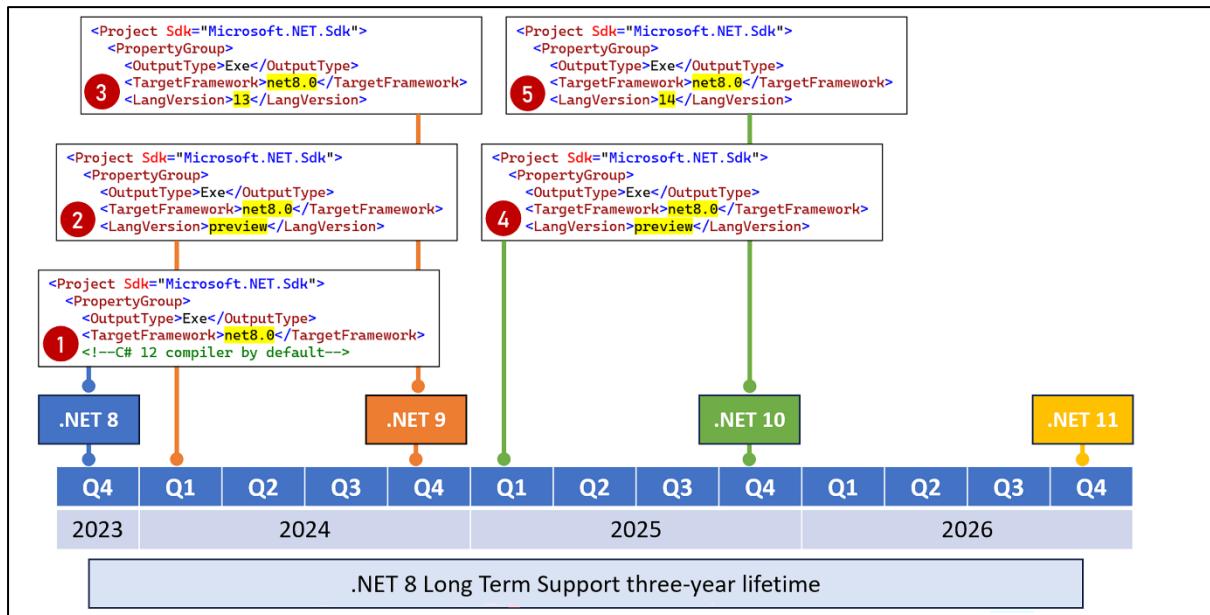
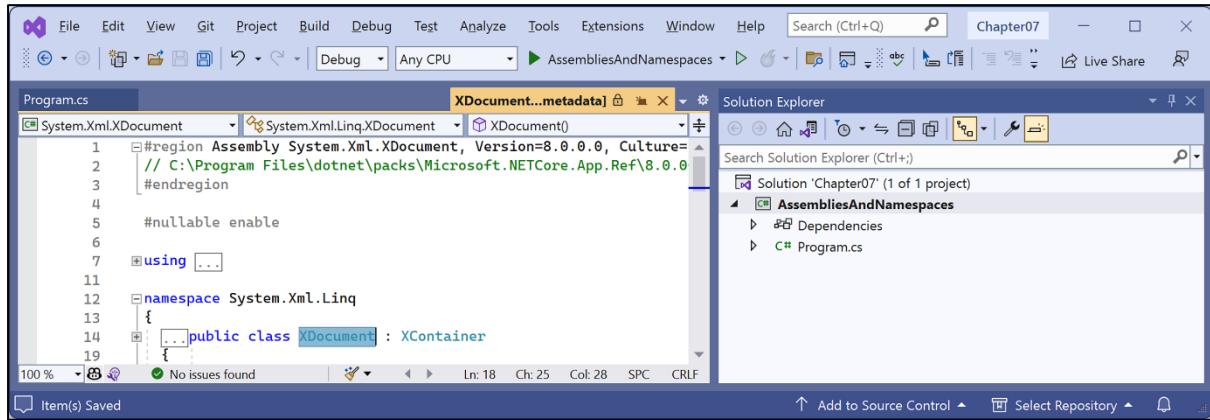
Chapter 6: Implementing Interfaces and Inheriting Classes







Chapter 7: Packaging and Distributing .NET Types



ILSpy

File View Window Help

(Default) C# C# 11.0 / VS 2022

Assemblies

- DotNetEverywhere (1.0.0.0, .NETCoreApp, v8.0)
 - Metadata
 - Debug Metadata (From portable PDB)
 - References
 - {} (Module)
 - Program
 - Base Types
 - Derived Types
 - Program()
 - <Main>\$({string[]}) : void
- System.Runtime (8.0.0.0, .NETCoreApp, v8.0)
- System.Console (8.0.0.0, .NETCoreApp, v8.0)
- System.Private.CoreLib (8.0.0.0, .NETCoreApp, v8.0)

```

<Main>$({string[]}) : void
// DotNetEverywhere, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// Program
using System;

private static void <Main>$({string[]} args)
{
    Console.WriteLine("I can run everywhere!");
    Console.WriteLine($"OS Version is {Environment.OSVersion}.");
    if (OperatingSystem.IsMacOS())
    {
        Console.WriteLine("I am macOS.");
    }
    else if (OperatingSystem.IsWindowsVersionAtLeast(10, 0, 22000))
    {
        Console.WriteLine("I am Windows 11.");
    }
}

```

ILSpy

File View Window Help

(Default) C# C# 11.0 / VS 2022

Assemblies

- Count<TSource>(this IEnumerable<TSource>) : int
 - Chunk<TSource>(this IEnumerable<TSource>, int) : IEnum
 - ChunkIterator<TSource>(IEnumerable<TSource>, int) : IEn
 - Concat<TSource>(this IEnumerable<TSource>, IEnumerab
 - Contains<TSource>(this IEnumerable<TSource>, TSource)
 - Contains<TSource>(this IEnumerable<TSource>, TSource, IEqualityG
 - Count<TSource>(this IEnumerable<TSource>) : int
 - Count<TSource>(this IEnumerable<TSource>, Func<TSource, bool> predicate) : int
 - CreateSelectPartitionIterator<TResult,TSource>(Func<TSource, TResult> selector, IEqualityComparer<TSource> equalityComparer) : IEn
 - DefaultIfEmpty<TSource>(this IEnumerable<TSource>) : IEn
 - Distinct<TSource>(this IEnumerable<TSource>) : IEnumerab
 - Distinct<TSource>(this IEnumerable<TSource>, IEqualityComparer<TSource>) : IEn
 - DistinctBy<TSource,TKey>(this IEnumerable<TSource>, Func<TSource, TKey>) : IEn
 - DistinctBy<TSource,TKey>(this IEnumerable<TSource>, Func<TSource, TKey>, IEqualityComparer<TKey>) : IEn
 - DistinctByLIterator<TSource,TKey>(IList<TSource>, Func<TSource, TKey>, IEqualityComparer<TKey>) : IEn
 - ElementAt<TSource>(this IEnumerable<TSource>, int) : TS
 - ElementAt<TSource>(this IEnumerable<TSource>, Index) : TSource
 - ElementOrDefault<TSource>(this IEnumerable<TSource>) : TSource
 - ElementOrDefault<TSource>(this IEnumerable<TSource>, TSource) : TSource
 - Empty<TResult>() : IEnumerab
 - Except<TSource>(this IEnumerable<TSource>, IEnumerabl
 - ExceptBy<TSource,TKey>(this IEnumerable<TSource>, IEn
 - ExceptBy<TSource,TKey>(this IEnumerable<TSource>, IEqualityComparer<TKey>) : IEn
 - ExceptByLIterator<TSource,TKey>(IList<TSource>, Func<TSource, TKey>, IEqualityComparer<TKey>) : IEn
 - First<TSource>(this IEnumerable<TSource>) : TSource
 - First<TSource>(this IEnumerable<TSource>, Func<TSource, bool> predicate) : TSource
 - FirstOrDefault<TSource>(this IEnumerable<TSource>) : TS

```

Count<TSource>(this IEnumerable<TSource>) : int
// System.Linq, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a
// System.Linq.Enumerable
+ using ...

public static int Count<TSource>(this IEnumerable<TSource> source)
{
    if (source == null)
    {
        ThrowHelper.ThrowArgumentNullException(ExceptionArgument.source);
    }
    if (source is ICollection<TSource> collection)
    {
        return collection.Count;
    }
    if (source is IListProvider<TSource> i IListProvider)
    {
        return i IListProvider.GetCount(onlyIfCheap: false);
    }
    if (source is ICollection collection2)
    {
        return collection2.Count;
    }
    int num = 0;
    using IEnumerator<TSource> enumerator = source.GetEnumerator();
    while (enumerator.MoveNext())
    {
        num = checked(num + 1);
    }
    return num;
}

```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Chapter07

Program.cs

System.Linq (8.0.0.0)

```

1 // Licensed to the .NET Foundation under one or more agree
2 // The .NET Foundation licenses this file to you under the
3
4 using System.Collections;
5 using System.Collections.Generic;
6
7 namespace System.Linq
8 {
9     public static partial class Enumerable
10    {
11        public static int Count<TSource>(this IEnumerable<TSource> source)
12        {
13            if (source == null)
14            {
15                ThrowHelper.ThrowArgumentNullException(ExceptionArgument.source);
16            }
17        }
18    }
19 }

```

Count.cs [SourceLink]

Count<TSource>(IEnumerable<TSource> source)

Count<TSource>(IEnumerable<TSource> source, Func<TSource, bool> predicate)

LongCount<TSource>(IEnumerable<TSource> source)

LongCount<TSource>(IEnumerable<TSource> source, Func<TSource, bool> predicate)

TryGetNonEnumeratedCount<TSource>(IEnumerable<TSource> source, out int count)

100% No issues found Error List Output Ready Add to Source Control Select Repository

NuGet Gallery | Packt.CSdotnet.S

https://www.nuget.org/packages/Packt.CSdotnet.SharedLibrary/8.0.0

nuget Packages Upload Statistics Documentation Downloads Blog markjprice

Search for packages...

You successfully uploaded Packt.CSdotnet.SharedLibrary 8.0.0.

Packt.CSdotnet.SharedLibrary 8.0.0

.NET Standard 2.0

This package has not been published yet. It will appear in search results and will be available for install/restore after both validation and indexing are complete. Package validation and indexing may take up to an hour. [Read more](#)

The package icon will become available after this package is indexed.

Downloads Full stats →

Total 2.9K

Current version 0

Per day average 2

About

Last updated a few seconds ago

NuGet Gallery | Packt.CSdotnet.S

https://www.nuget.org/packages/Packt.CSdotnet.SharedLibrary/8.0.0#supportedframeworks

README Frameworks Dependencies Used By Versions

Release Notes

Product Versions

Product	Versions
.NET	net5.0 net5.0-windows net6.0 net6.0-android net6.0-ios net6.0-maccatalyst net6.0-macos net6.0-tvos net6.0-windows net7.0 net7.0-android net7.0-ios net7.0-maccatalyst net7.0-macos net7.0-tvos net7.0-windows
.NET Core	netcoreapp2.0 netcoreapp2.1 netcoreapp2.2 netcoreapp3.0 netcoreapp3.1
.NET Standard	netstandard2.0 netstandard2.1
.NET Framework	net461 net462 net463 net47 net471 net472 net48 net481

Download package (24.92 KB)

Open in NuGet Package Explorer

Open in FuGet Package Explorer

Manage

Manage package

Contact support

Owners Contact owners →

markjprice

string extensions
packt csharp dotnet

Packt.CSdotnet.SharedLibrary 8.0 X +

https://nuget.info/packages/Packt.CSdotnet.SharedLibrary/8.0.0

File

Metadata Dependencies

Id: Packt.CSdotnet.SharedLibrary
Version: 8.0.0
[View Metadata source](#)

Digital Signatures

Repository: NuGet.org Repository by Microsoft
@ Tuesday, 11 July 2023 17:06:47
by DigiCert Timestamp 2022 - 2
Owners: markjprice
Service Index: <https://api.nuget.org/v3/>

Health

POWERED BY  UNO®
PLATFORM

Contents

- lib
- netstandard2.0
 - SharedLibrary.dll
 - packt-csdotnet-sharedlibrary.png
 - readme.md

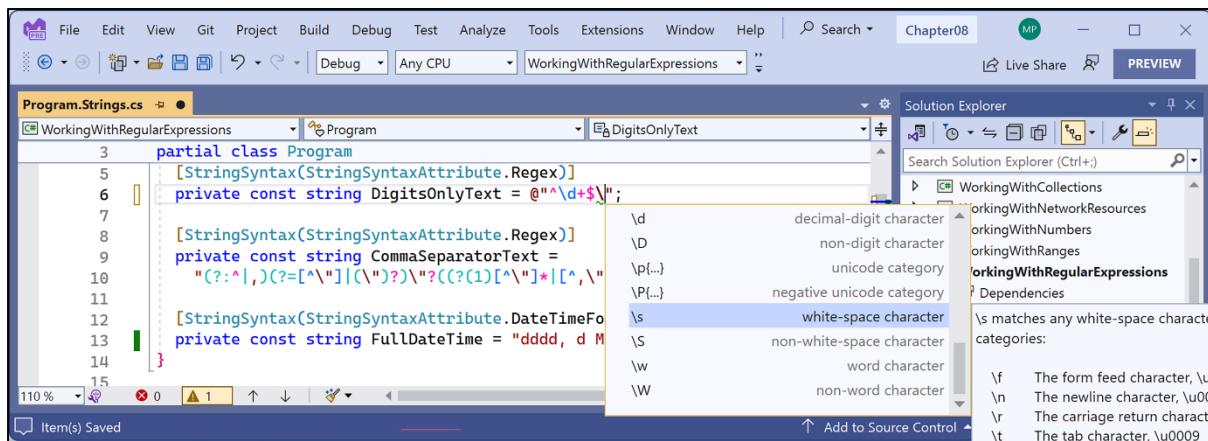
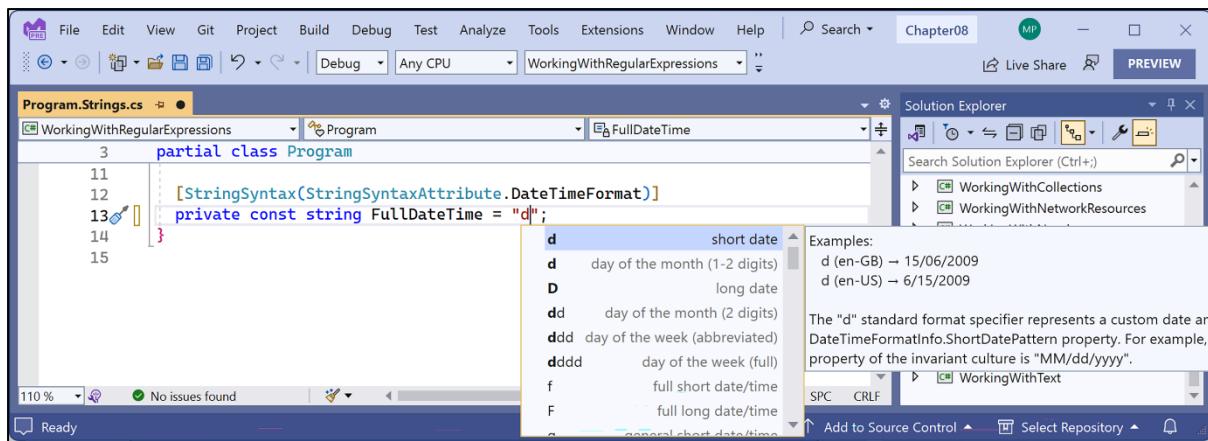
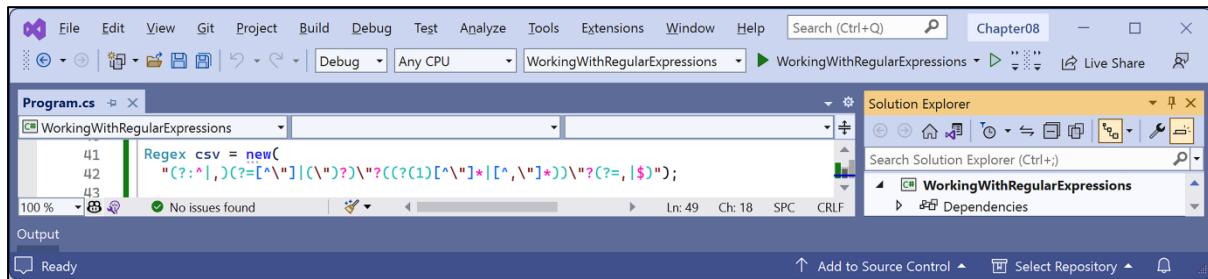
SharedLibrary.dll

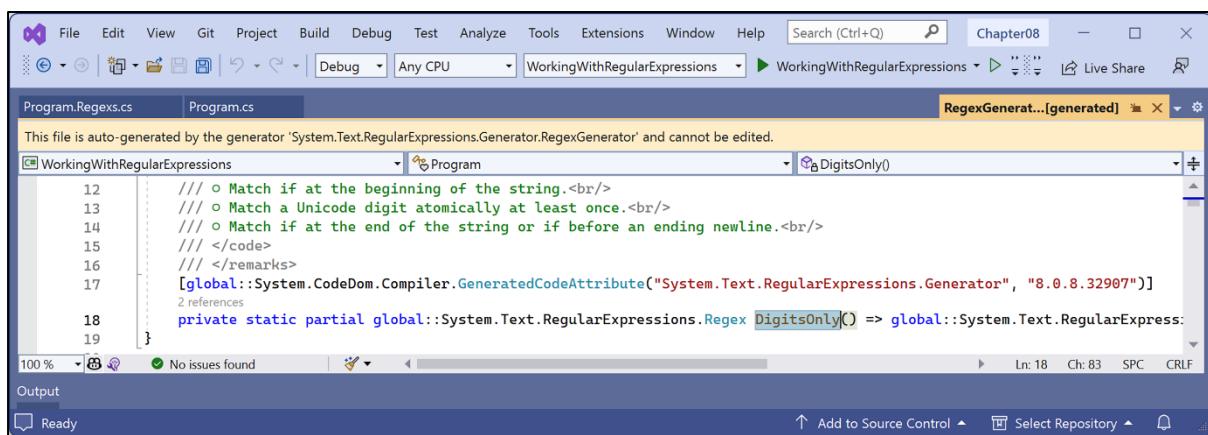
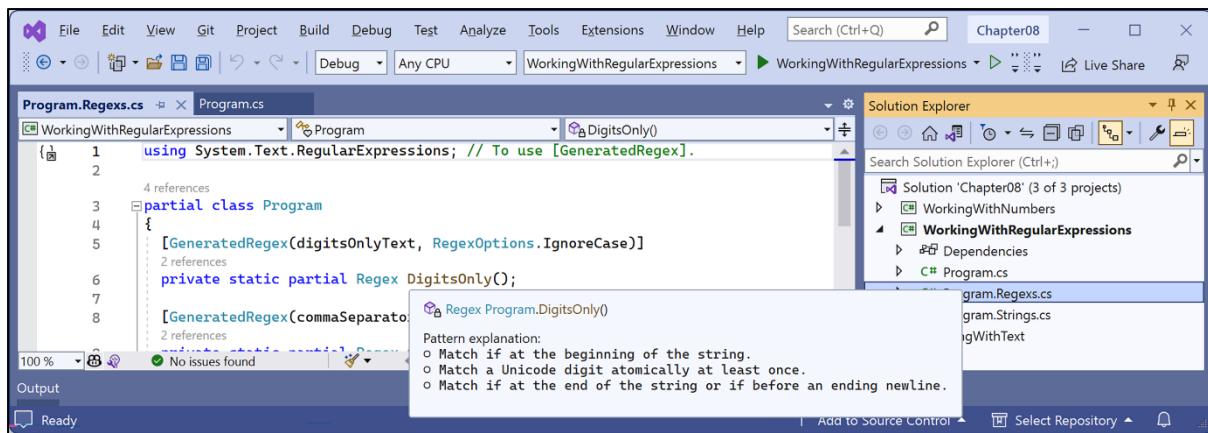
Size: 5.5 KB

Assembly Attributes PDB Info PDB Metadata References PDB Sources

Full Name: SharedLibrary, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
Strong Name: No
AssemblyCompany: Mark J Price
AssemblyConfiguration: Release
AssemblyCopyright: Copyright © 2016-2023 Packt Publishing Limited

Chapter 8: Working with Common .NET Types





Chapter 9: Working with Files, Streams, and Serialization

Microsoft Visual Studio Debug Console

```
*** Handling cross-platform environments and filesystems ***
```

MEMBER	VALUE
Path.PathSeparator	:
Path.DirectorySeparatorChar	\
Directory.GetCurrentDirectory()	C:\cs12dotnet8\Chapter09\WorkingWithFileSystems\bin\Debug\net8.0
Environment.CurrentDirectory	C:\cs12dotnet8\Chapter09\WorkingWithFileSystems\bin\Debug\net8.0
Environment.SystemDirectory	C:\WINDOWS\system32
Path.GetTempPath()	C:\Users\markj\AppData\Local\Temp\
GetFolderPath(SpecialFolder .System)	C:\WINDOWS\system32
.ApplicationData)	C:\Users\markj\AppData\Roaming
.MyDocuments)	C:\Users\markj\OneDrive\Documents
.Personal)	C:\Users\markj\OneDrive\Documents

Terminal Shell Edit View Window Help

WorkingWithFileSystems -- zsh -- 109x20

```
[markjprice@Marks-Mac-mini WorkingWithFileSystems % dotnet run
```

```
*** Handling cross-platform environments and filesystems ***
```

MEMBER	VALUE
Path.PathSeparator	:
Path.DirectorySeparatorChar	/
Directory.GetCurrentDirectory()	/Users/markjprice/cs12dotnet8/Chapter09/WorkingWithFileSystems
Environment.CurrentDirectory	/Users/markjprice/cs12dotnet8/Chapter09/WorkingWithFileSystems
Environment.SystemDirectory	/System
Path.GetTempPath()	/var/folders/6b/grhsfcxsx75qb8sf9hf3r1qgm0000gn/T/
GetFolderPath(SpecialFolder .System)	/System
.ApplicationData)	/Users/markjprice/Library/Application Support
.MyDocuments)	/Users/markjprice/Documents
.Personal)	/Users/markjprice/Documents

Microsoft Visual Studio Debug Console

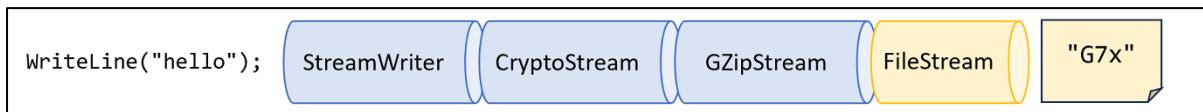
*** Managing drives ***

NAME	TYPE	FORMAT	SIZE (BYTES)	FREE SPACE
C:\	Fixed	NTFS	510,759,268,352	148,998,021,120

WorkingWithFileSystems — -zsh — 101x16

*** Managing drives ***

NAME	TYPE	FORMAT	SIZE (BYTES)	FREE SPACE
/	Fixed	apfs	494,384,795,648	226,914,848,768
/dev	Ram	devfs	206,848	0
/System/Volumes/VM	Fixed	apfs	494,384,795,648	226,914,848,768
/System/Volumes/Preboot	Fixed	apfs	494,384,795,648	226,914,848,768
/System/Volumes/Update	Fixed	apfs	494,384,795,648	226,914,848,768
/System/Volumes/xarts	Fixed	apfs	524,288,000	506,101,760
/System/Volumes/iSCPreboot	Fixed	apfs	524,288,000	506,101,760
/System/Volumes/Hardware	Fixed	apfs	524,288,000	506,101,760
/System/Volumes/Data	Fixed	apfs	494,384,795,648	226,914,848,768
/System/Volumes/Data/home	Network	autofs	0	0
/Volumes/LaCie	Fixed	hfs	4,000,443,056,128	875,921,227,776



← Thread

James Newton-King @JamesNK

Negative 2 billion downloads in NuGet Package Explorer 😅

cc @NuGetPE

Select Package

Search (Ctrl+E) Show pre-release packages

Newtonsoft.Json by James Newton-King -2.1G downloads v13.0.2-beta1

Newtonsoft.Json is a popular high-performance JSON framework for .NET.

Microsoft.Extensions.DependencyInjection by Microsoft, 1.4G dow v7.0.0-preview.7.22375.6

Microsoft.Extensions.DependencyInjection is the default implementation of dependency injection for Microsoft.Extensions.DependencyInjection.

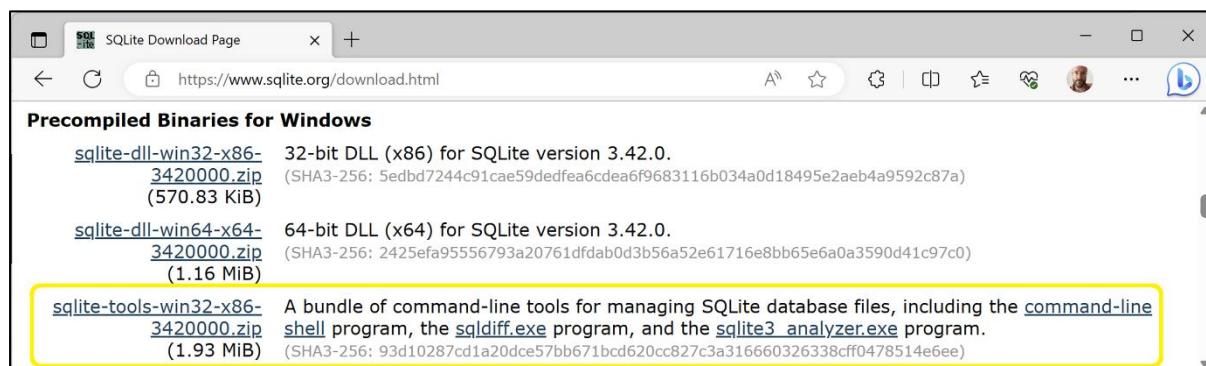
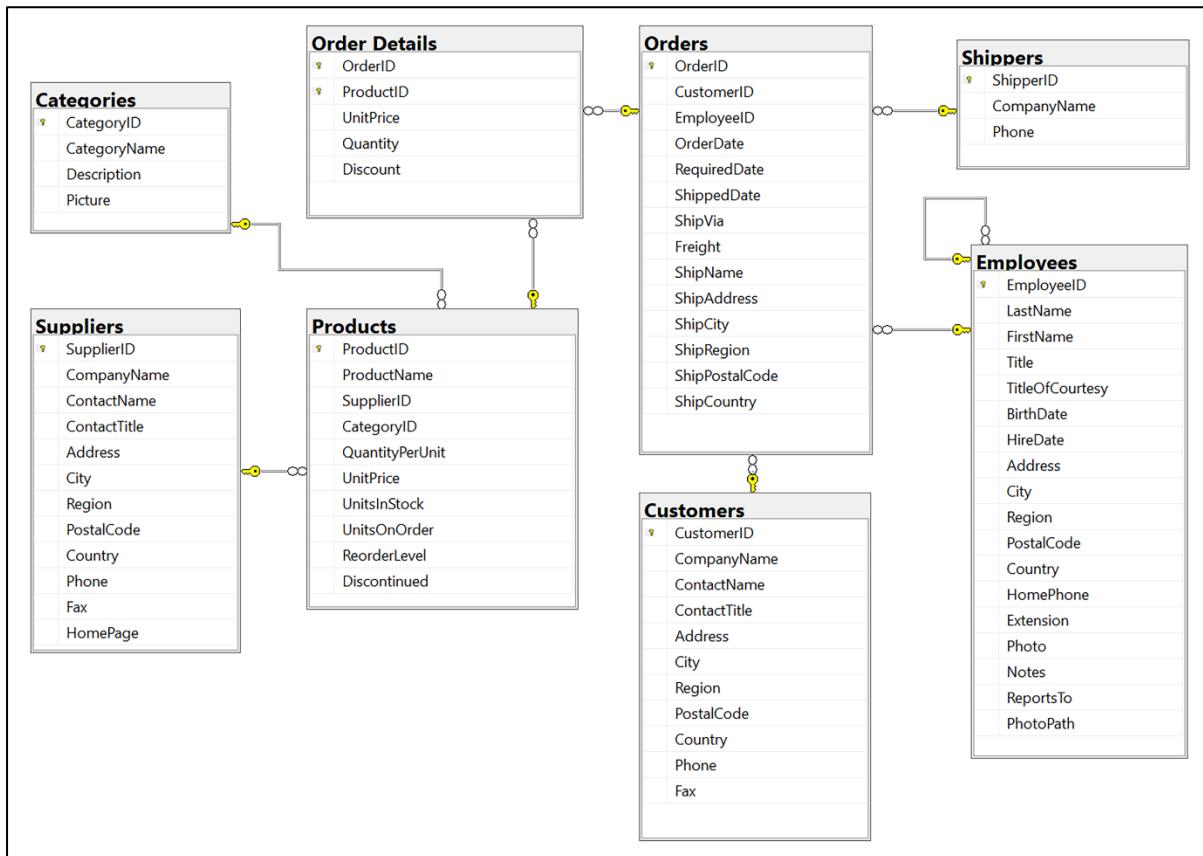
JamesNK commented on 29 Oct 2018

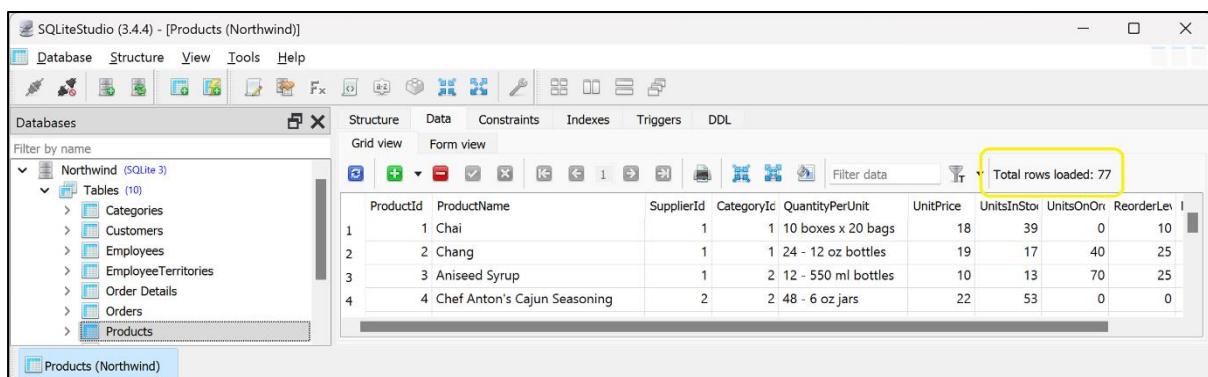
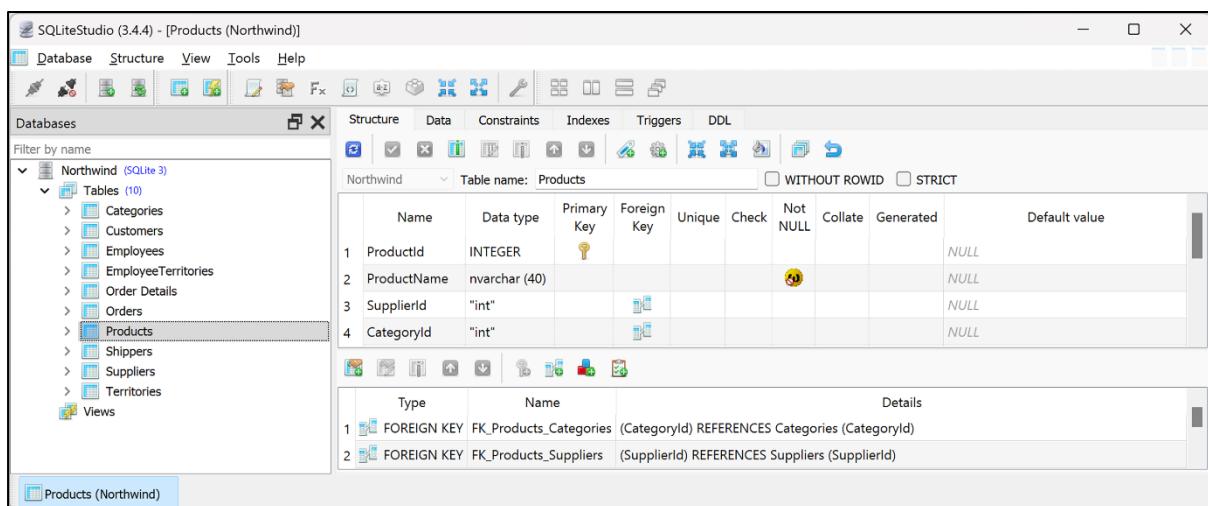
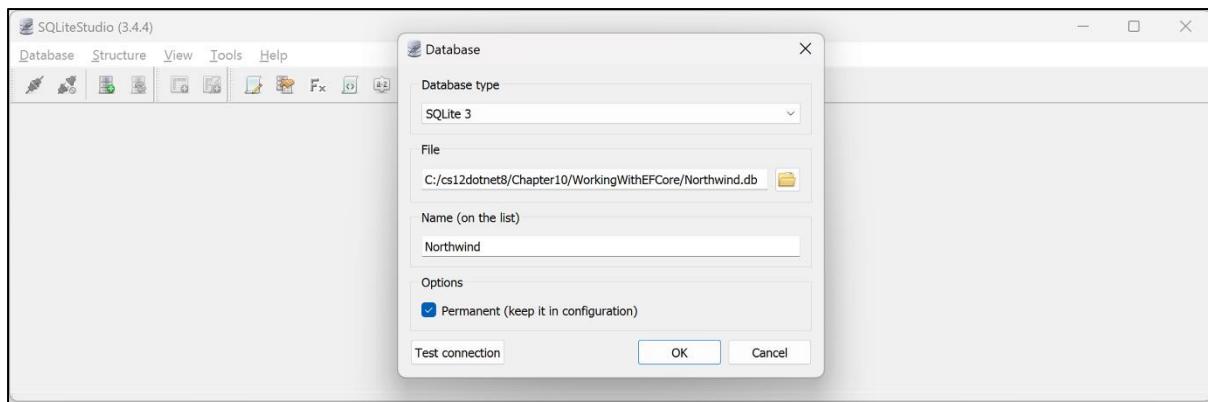
Member + ...

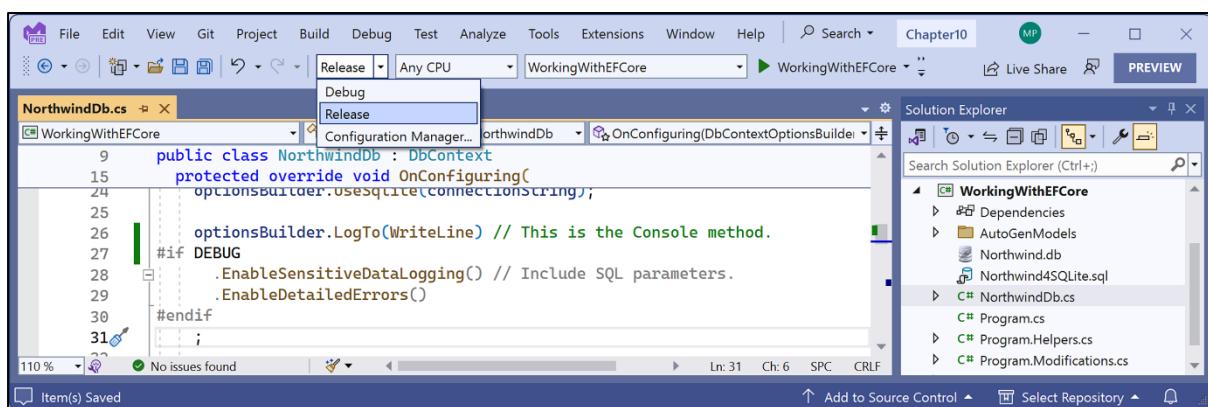
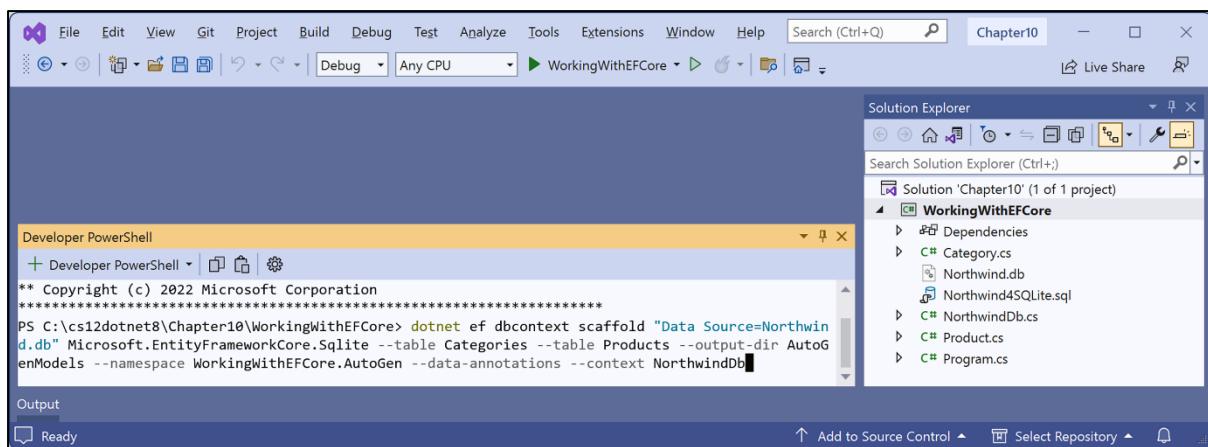
@Thorium Json.NET isn't going away. You aren't losing anything. This is another option for simple and high performance scenarios.

19

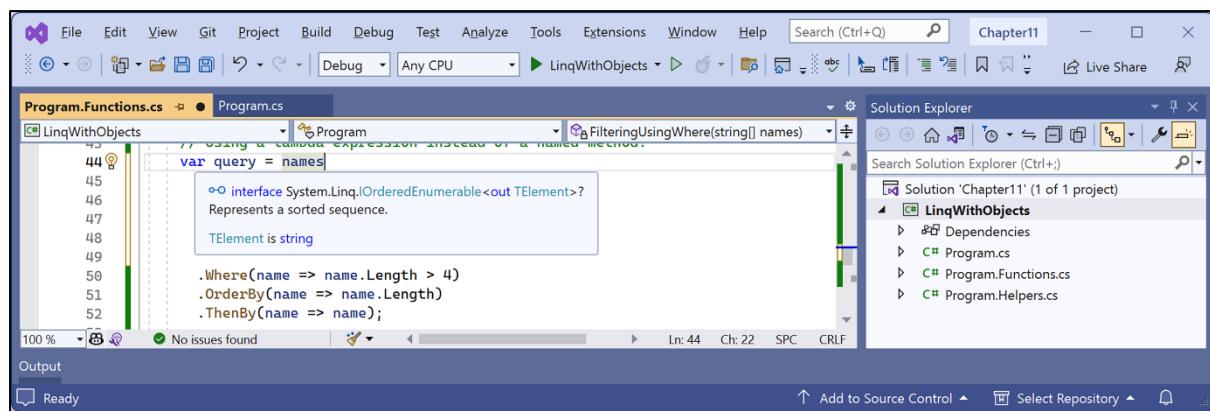
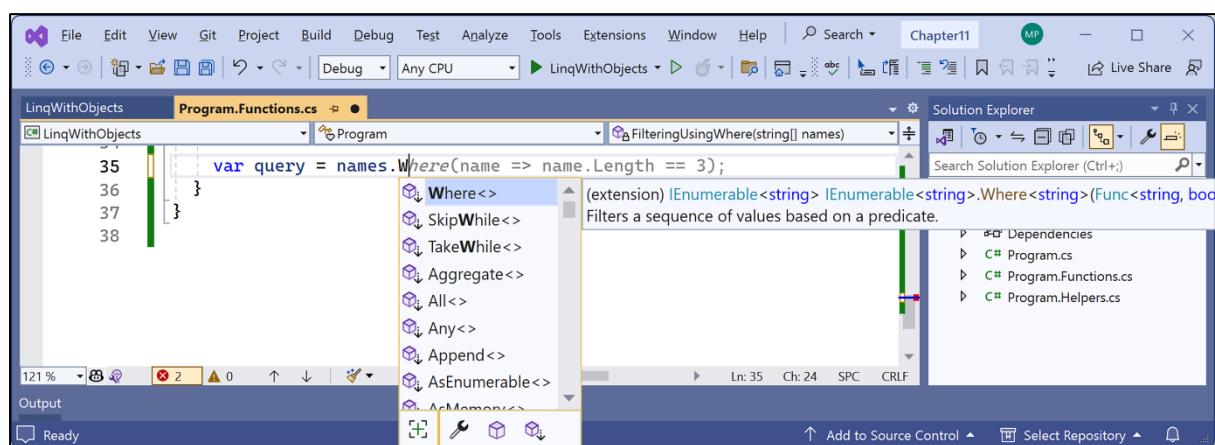
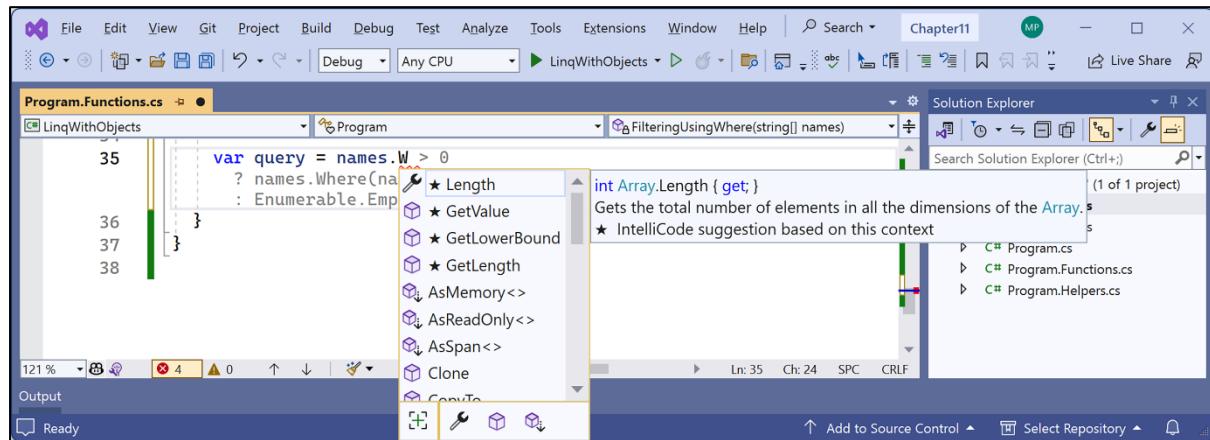
Chapter 10: Working with Data Using Entity Framework Core

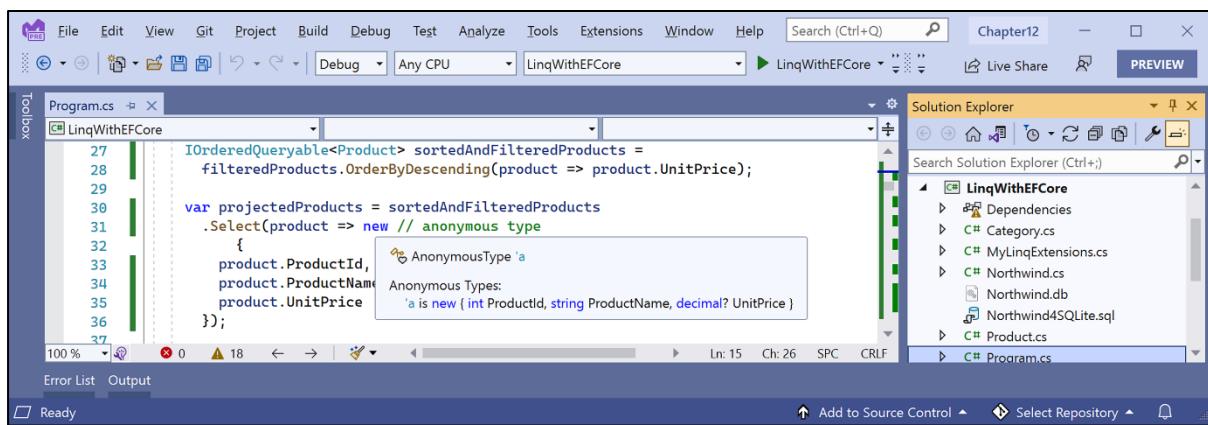
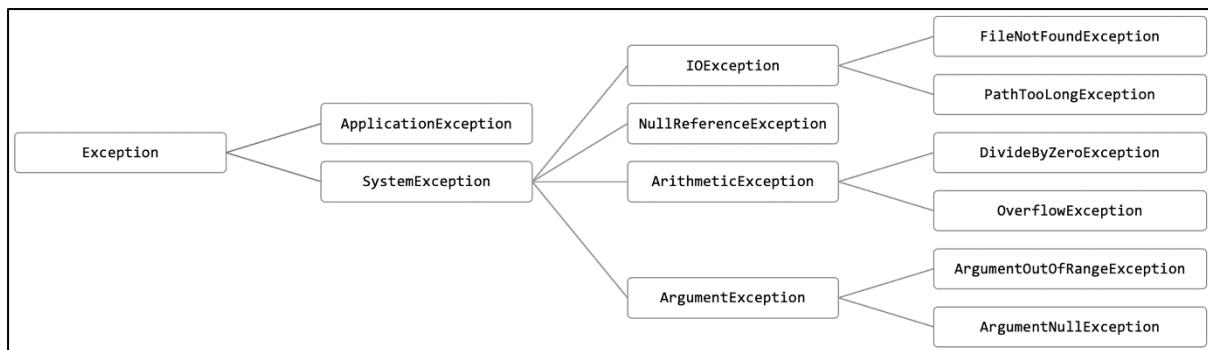






Chapter 11: Querying and Manipulating Data Using LINQ





 **Amichai Mantinband**
@amantinband

Surprised by the number of wrong answers on the youtube poll! #dotnet twitter for the save?

```

IQueryable<Task> tasks = Enumerable.Range(0, 2)
    .Select(_ => Task.Run(() => Console.WriteLine("")));
  
```

await Task.WhenAll(tasks);
Console.WriteLine(\$"{tasks.Count()} stars!");

**2 stars!	49.3%
2 stars!	14.5%
****2 stars!	15.9%
Something else 🎉!	20.3%

207 votes · Final results
3:39 PM · Aug 15, 2022 · Twitter Web App

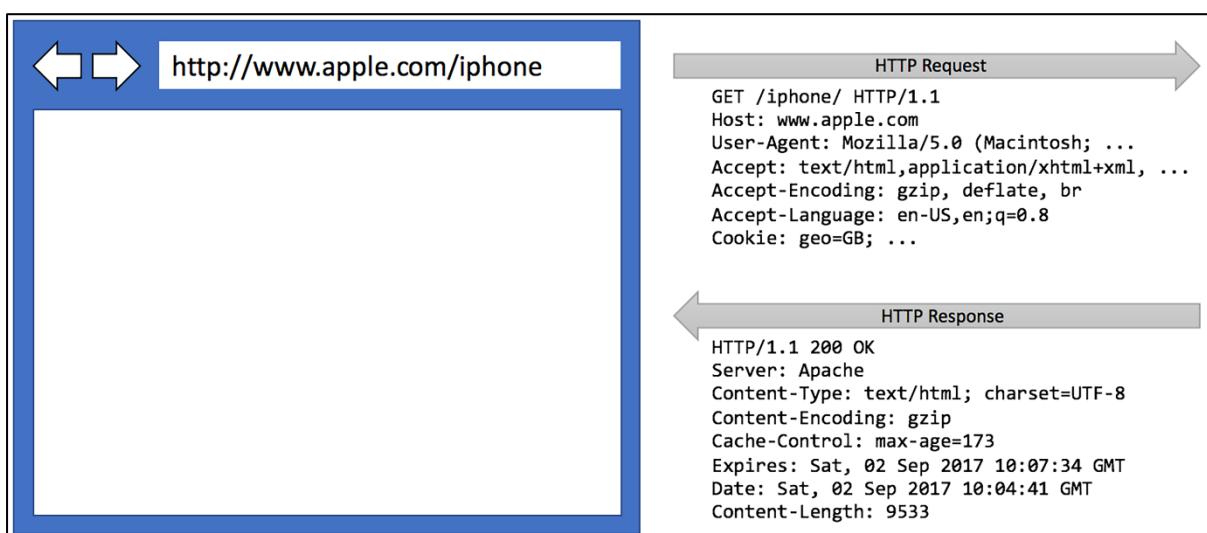
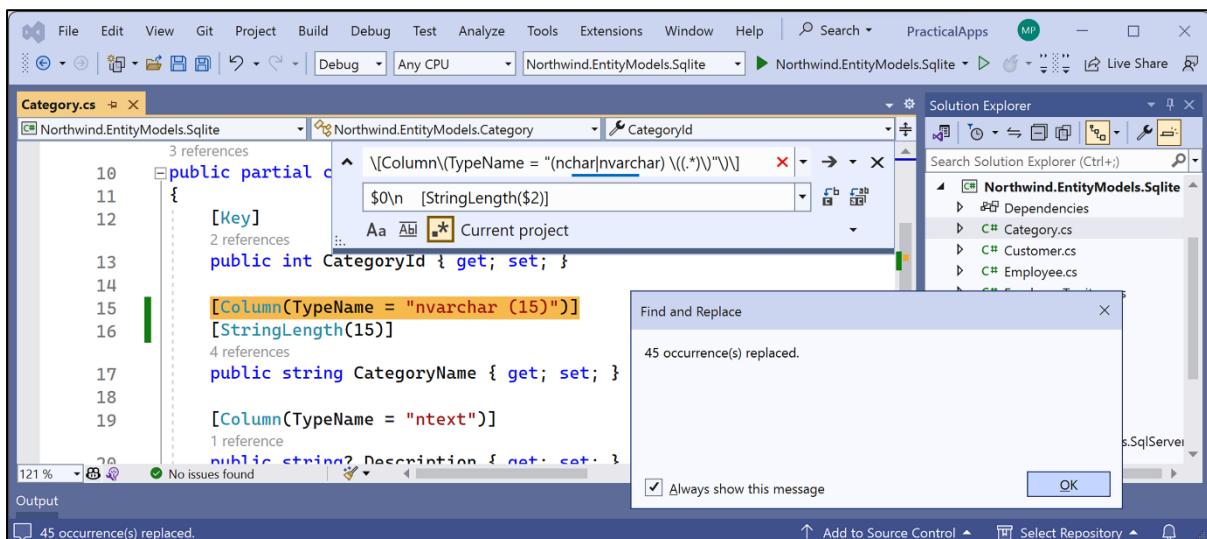
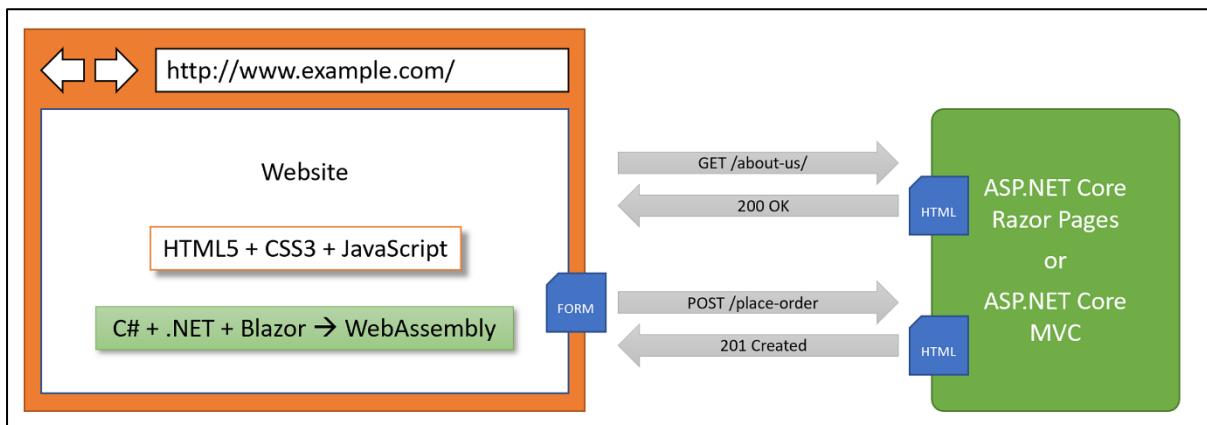
 **Amichai Mantinband**
254 Tweets

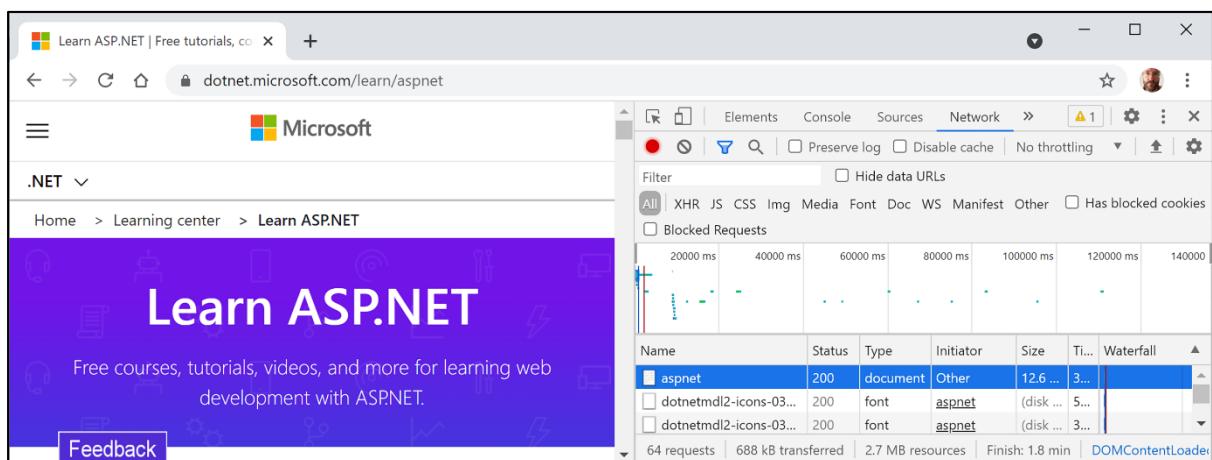
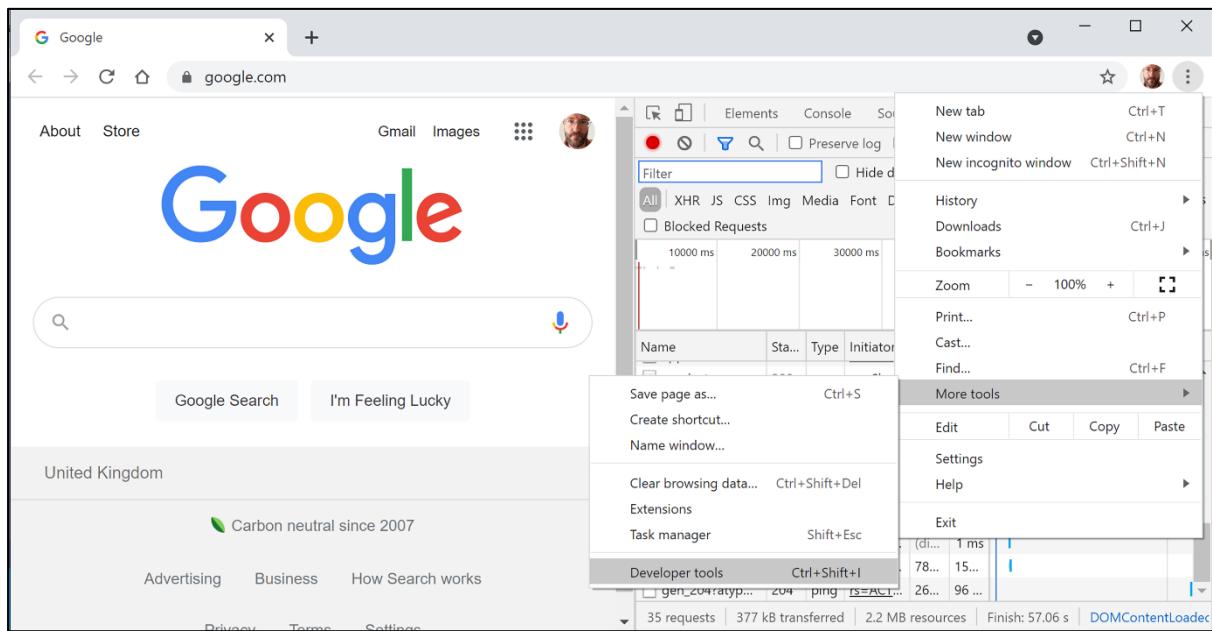


 **Amichai Mantinband**
@amantinband

Software engineer @Microsoft | Dad | Amazing husband according to me | Average husband according to wife
Joined October 2021
401 Following 1,343 Followers

Chapter 12: Introducing Web Development Using ASP.NET Core





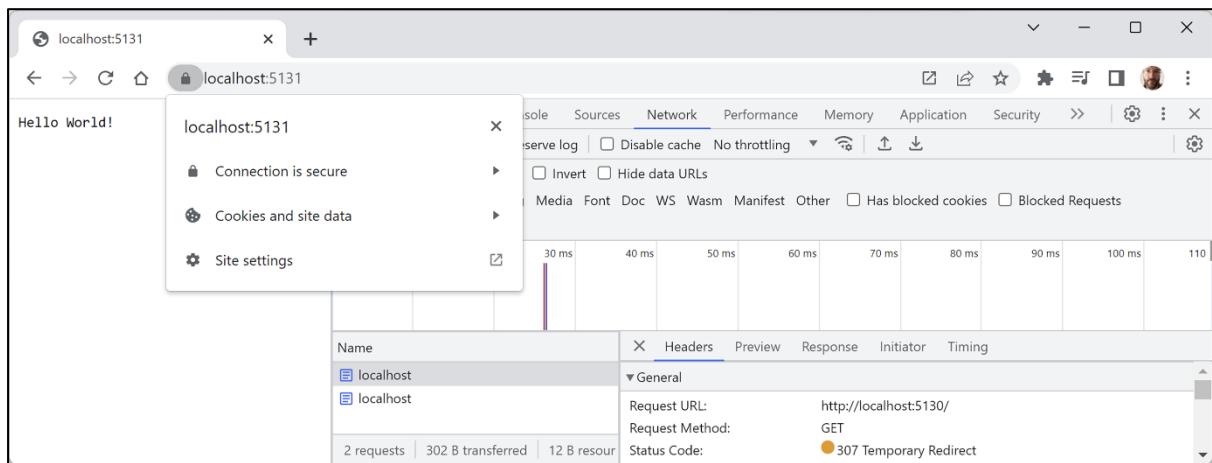
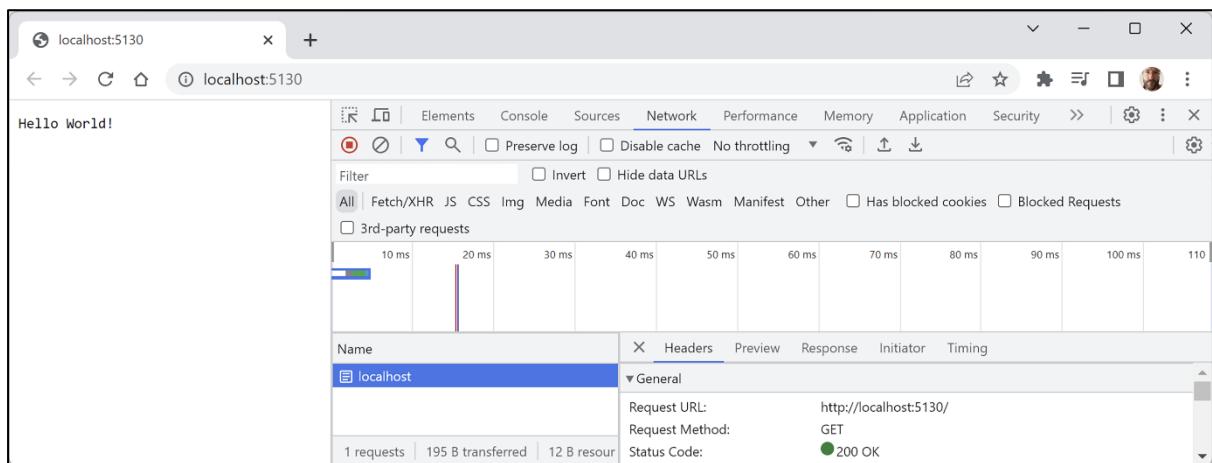
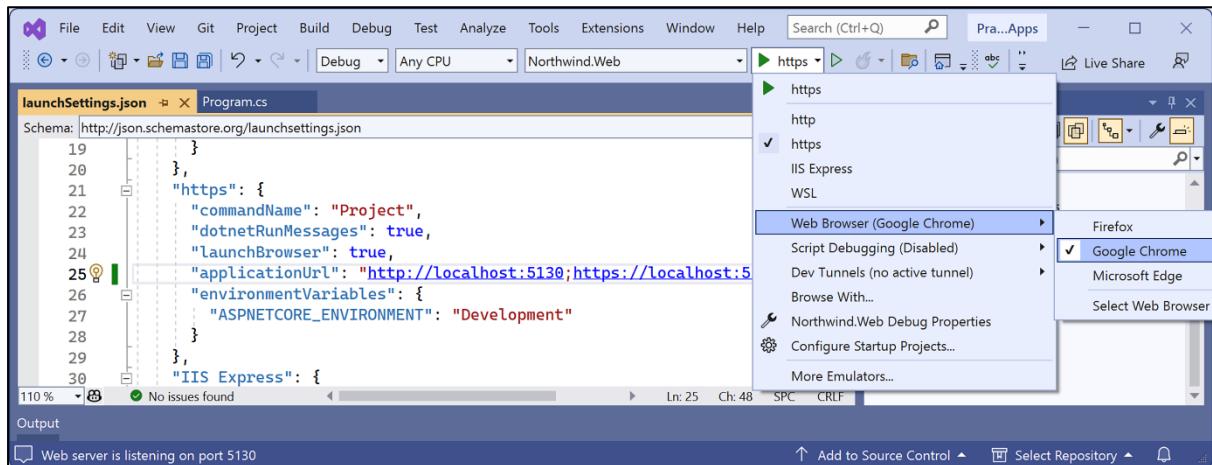
Screenshot of a browser developer tools Network tab showing a request to dotnet.microsoft.com/learn/aspnet.

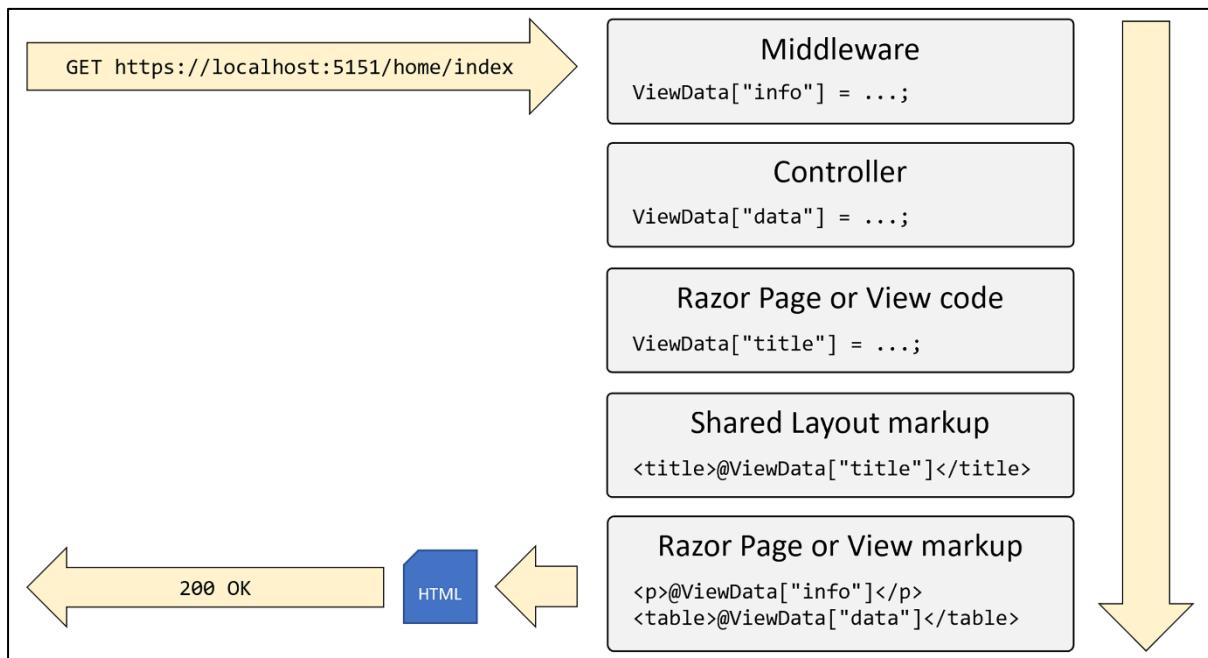
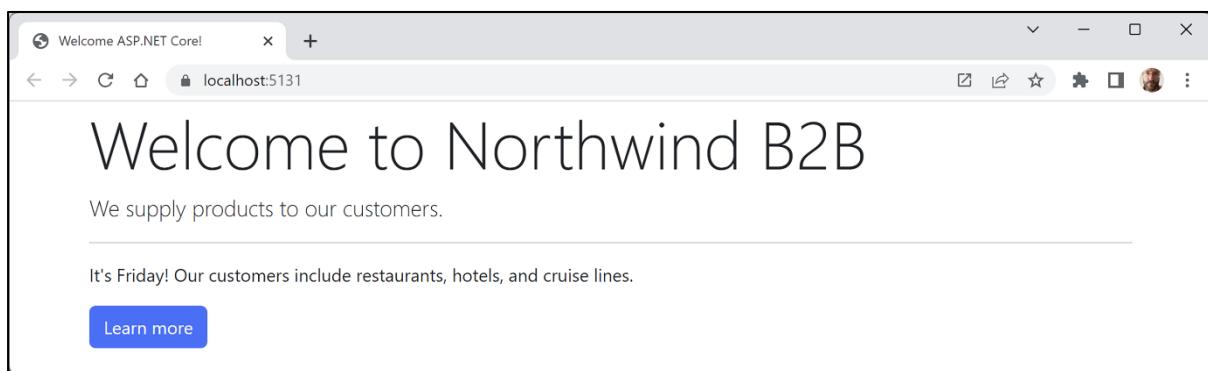
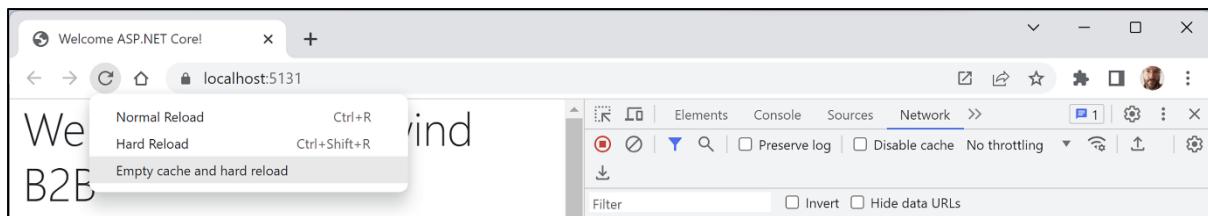
The Network tab displays the following details:

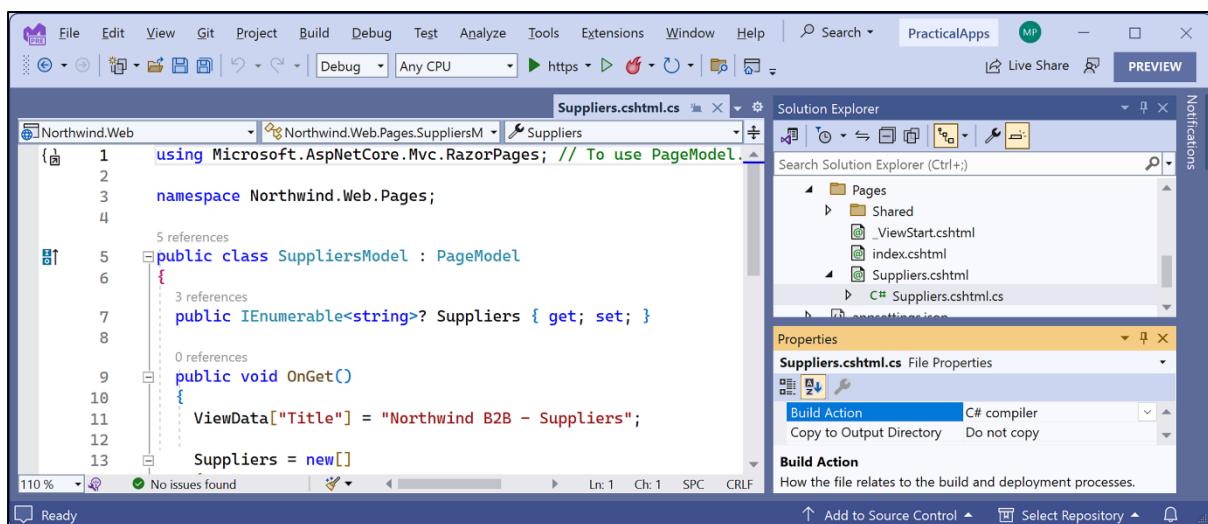
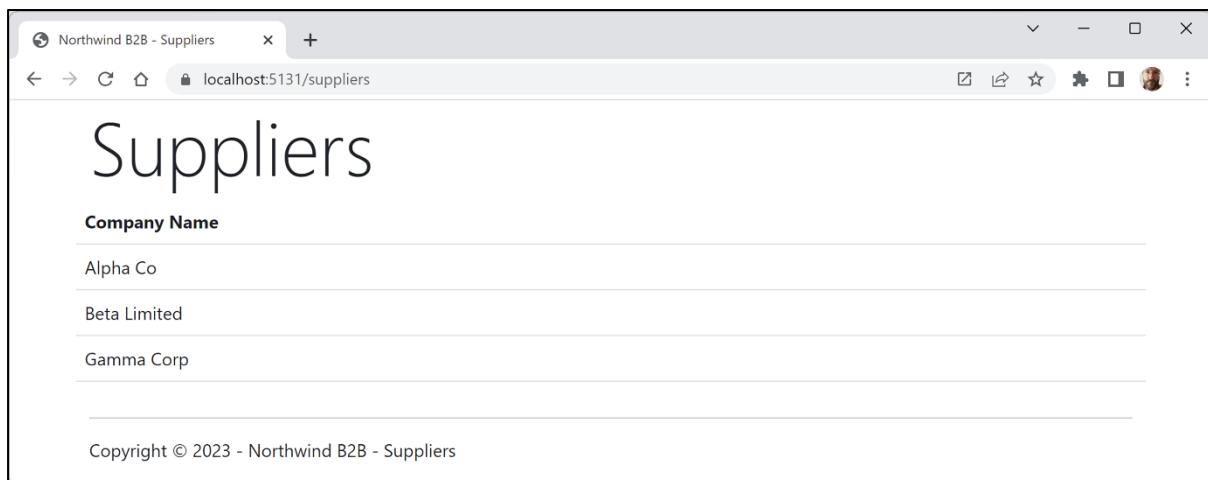
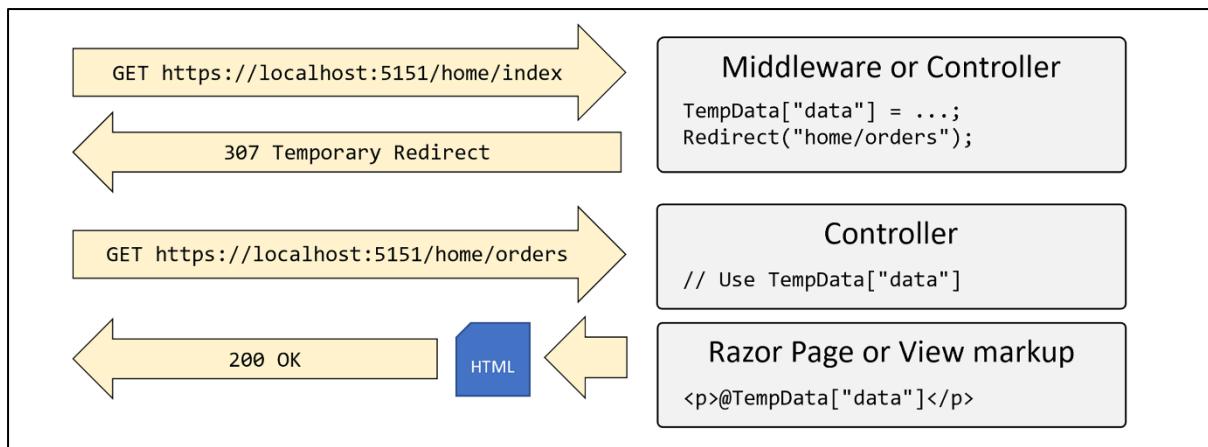
- Request URL:** <https://dotnet.microsoft.com/learn/aspnet>
- Request Method:** GET
- Status Code:** 200
- Remote Address:** [2620:1ec:bdf:64]:443
- Referrer Policy:** strict-origin-when-cross-origin
- Headers:**
 - :authority: dotnet.microsoft.com
 - :method: GET
 - :path: /learn/aspnet
 - :scheme: https
 - accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
 - accept-encoding: gzip, deflate, br
 - accept-language: en-US,en;q=0.9,sv;q=0.8
- Cookies:**
 - cookie: MC1=GUID=54a83491596e42eb9d82baf90b40c0d4&iHASH=54a8&LV=202104&V=4&LU=1617522068644; mbox=session#8a46d478da9c43a5845df1d0e4fe86a7#1619507375|PC#8a46d478da9c43a5845df1d0e4fe86a7_37_0#1653692607; ai_user=v

The Network tab also shows a list of resources loaded, including files like aspnet, dotnetmdl2-icons-032021-2.v, and bootstrap-custom.min.css.

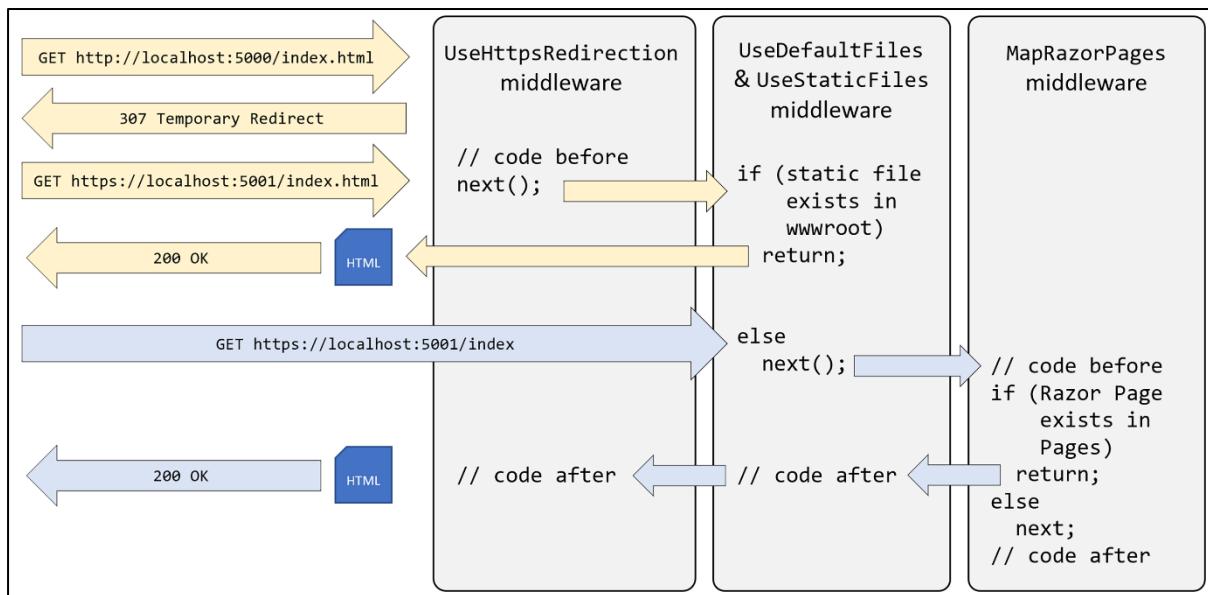
Chapter 13: Building Websites Using ASP.NET Core Razor Pages



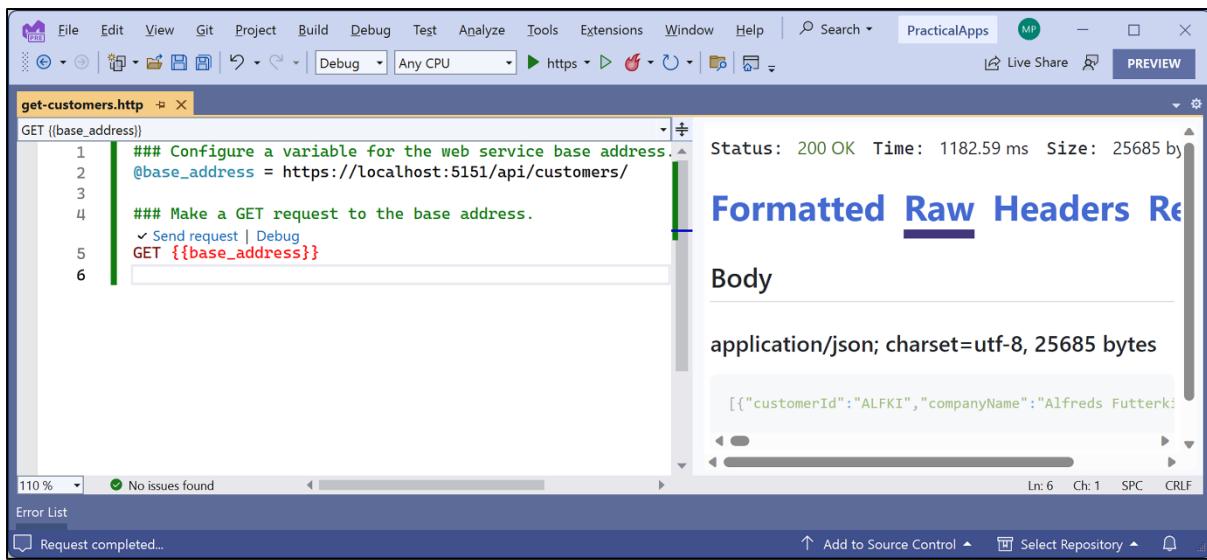
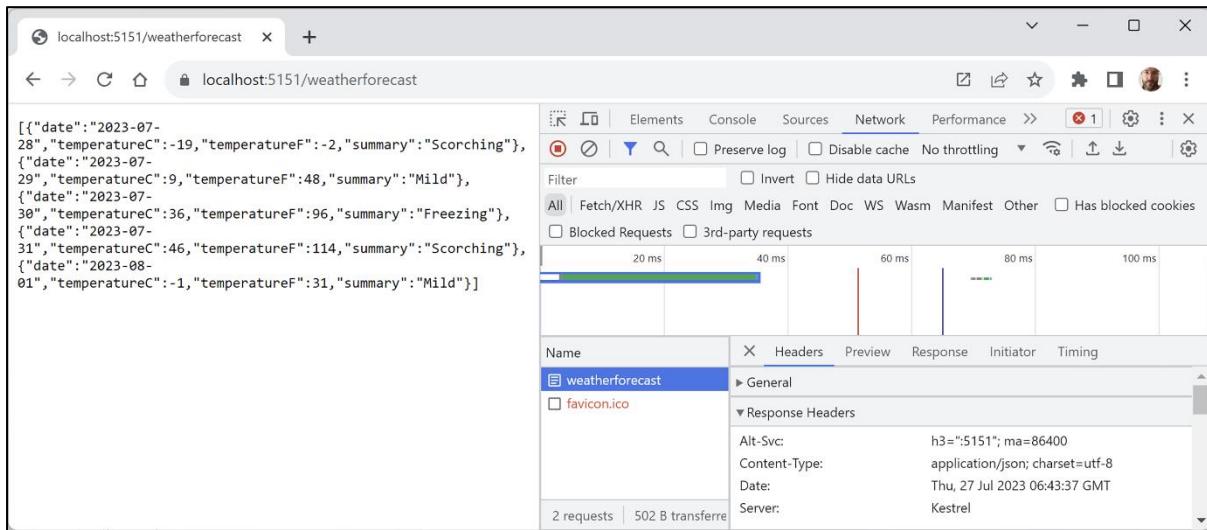




Company Name	Country	Phone
G'day, Mate	Australia	(02) 555-5914
Pavlova, Ltd.	Australia	(03) 444-2343
Refrescos Americanas LTDA	Brazil	(11) 555 4640
Forêts d'éables	Canada	(514) 555-2955



Chapter 14: Building and Consuming Web Services



File Edit Selection View Go Run Terminal Help get-customers.http - Visual Studio Code

get-customers.http

```
10  ## Get customers in USA in XML format
11  GET {{base_address}}?country=USA
12  Accept: application/xml
13
14  ## Get Alfre's Futterkiste
15  GET {{base_address}}ALFKI
16
17  ## Get a non-existent customer
18  GET {{base_address}}abcxy
19
```

Response(18ms)

```
1  HTTP/1.1 200 OK
2  Content-Length: 5055
3  Connection: close
4  Content-Type: application/xml; charset=utf-8
5  Date: Thu, 27 Jul 2023 10:32:57 GMT
6  Server: Kestrel
7  Alt-Svc: h3=":5151"; ma=86400
8
9  <ArrayOfCustomer
10  xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
11  xmlns="http://schemas.datacontract.org/2004/07/Northwind.Entities.Models">
12  <Customer>
13    <Address>2732 Baker Blvd.</Address>
14    <City>Eugene</City>
15    <CompanyName>Great Lakes Food Market</CompanyName>
```

0 18ms 5.09 KB [Azurite Queue Service] [Azurite Blob Service] Ln 10, Col 1 Spaces: 4 UTF-8 CRLF HTTP No Environment ⚙️ 🔍

File Edit Selection View Go Run Terminal Help Response(68ms) - Visual Studio Code

create-customer.http

```
1  ## Configure a variable for the web service base
2  @base_address = https://localhost:5151/api/customers
3
4  ## Make a POST request to the base address.
5  POST {{base_address}}
6  Content-Type: application/json
7
8  {
9    "customerId": "ABCXY",
10   " companyName": "ABC Corp",
11   " contactName": "John Smith",
12   " contactTitle": "Sir",
13   " address": "Main Street".
```

Response(68ms)

```
1  HTTP/1.1 201 Created
2  Connection: close
3  Content-Type: application/json; charset=utf-8
4  Date: Thu, 27 Jul 2023 10:51:23 GMT
5  Server: Kestrel
6  Alt-Svc: h3=":5151"; ma=86400
7  Location: https://localhost:5151/api/Customers/abcxy
8  Transfer-Encoding: chunked
9
10 <{
11   "customerId": "ABCXY",
12   " companyName": "ABC Corp",
13   " contactName": "John Smith",
```

0 68ms 448 B [Azurite Table Service] [Azurite Queue Service] [Azurite Blob Service] ⚙️ 🔍

The screenshot shows the Swagger UI interface for the Northwind Service API Version 1. At the top, there's a navigation bar with tabs for 'Select a definition' (set to 'Northwind Service API Version 1') and a dropdown menu. Below the header, the title 'Northwind.WebApi 1.0 OAS3' is displayed, along with a link to '/swagger/v1/swagger.json'. The main content area is organized into sections: 'Customers' and 'WeatherForecast'. The 'Customers' section contains five operations: GET /api/Customers (blue), POST /api/Customers (green), GET /api/Customers/{id} (light blue), PUT /api/Customers/{id} (orange), and DELETE /api/Customers/{id} (red). The 'WeatherForecast' section contains two operations: GET /WeatherForecast (blue) and GET /WeatherForecast/{days} (light blue). Each operation has a dropdown arrow to its right.

This screenshot shows a detailed view of the 'GET /api/Customers/{id}' operation from the 'Customers' section. It displays the 'Parameters' section. There is one parameter listed: 'id * required' of type 'string' (path), with the value 'alfki' entered into the input field. A 'Cancel' button is visible in the top right corner, and a large blue 'Execute' button is at the bottom.

This screenshot shows the results of executing the 'GET /api/Customers/alfki' request. The 'Request URL' is shown as 'https://localhost:5151/api/Customers/alfki'. Under the 'Server response' section, the status code '200' is listed under 'Code', and the 'Details' section shows the response body: a JSON object with fields 'customerId', 'companyName', and 'contactName', all set to their respective values for customer 'ALFKI'.

Request URL

```
https://localhost:5151/api/Customers/super
```

Server response

Code	Details
404	Error: response status is 404

Response body

```
{
  "type": "https://tools.ietf.org/html/rfc9110#section-15.5.5",
  "title": "Not Found",
  "status": 404,
  "traceId": "00-3202206d39b44ed303315fcebd25958d-53f424b03785c656-00"
}
```

Response headers

```
alt-svc: h3=":5151"; ma=86400
content-type: application/problem+json; charset=utf-8
date: Thu, 27 Jul 2023 11:18:18 GMT
server: Kestrel
```

[Copy](#) [Download](#)

Request URL

```
https://localhost:5151/api/Customers/bad
```

Server response

Code	Details
400	Error: response status is 400

Response body

```
{
  "type": "https://localhost:5151/customers/failed-to-delete",
  "title": "Customer ID bad found but failed to delete.",
  "status": 400,
  "detail": "More details like Company Name, Country and so on.",
  "instance": "/api/Customers/bad"
}
```

[Copy](#) [Download](#)

Solution 'PracticalApps' Property Pages

Configuration: N/A Platform: N/A Configuration Manager...

Current selection
 Single startup project Northwind.Mvc
 Multiple startup projects:

Project	Action
Northwind.DataContext.Sqlite	None
Northwind.WebApi	Start without debugging
Northwind.DataContext.SqlServer	None
Northwind.EntityModels.Sqlite	None
Northwind.EntityModels.SqlServer	None
Northwind.Mvc	Start without debugging
Northwind.Razor.Employees	None
Northwind.UnitTests	None
Northwind.Web	None

OK Cancel Apply

Solution Explorer

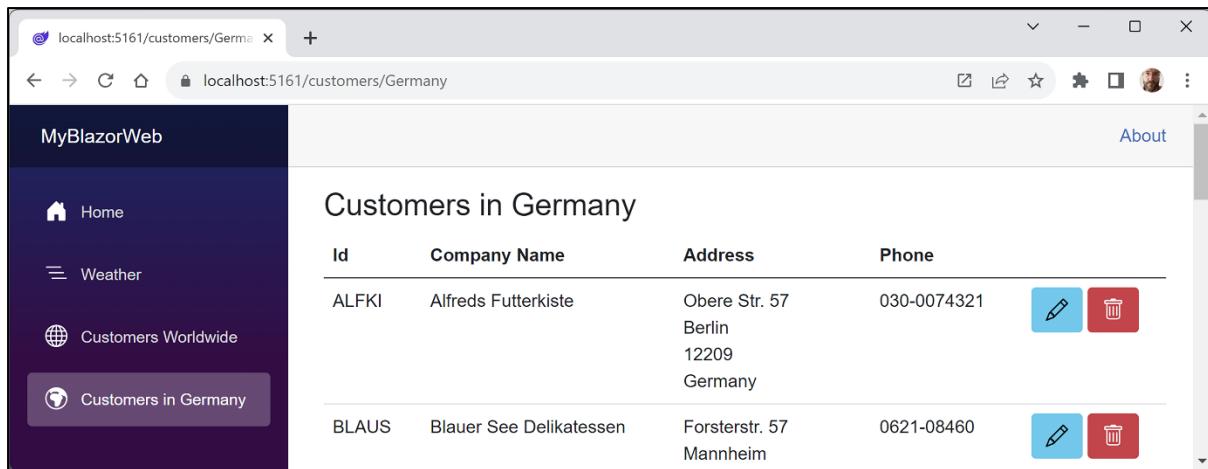
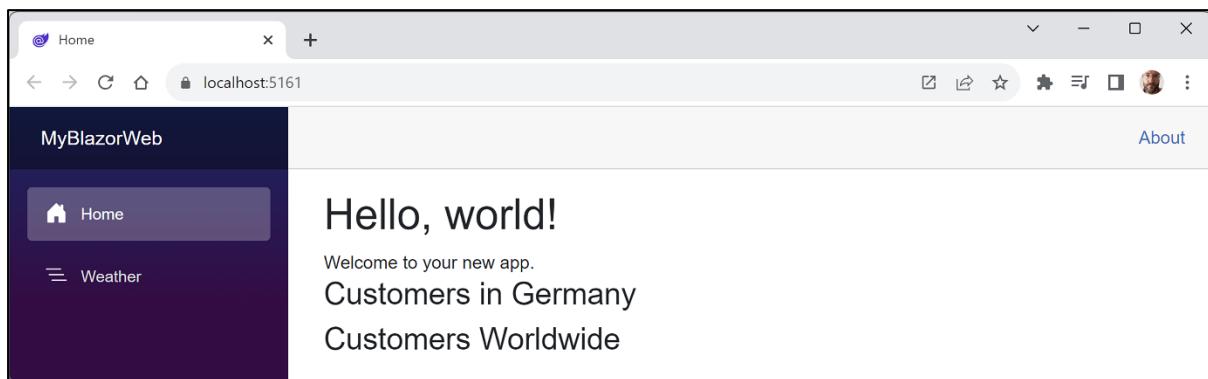
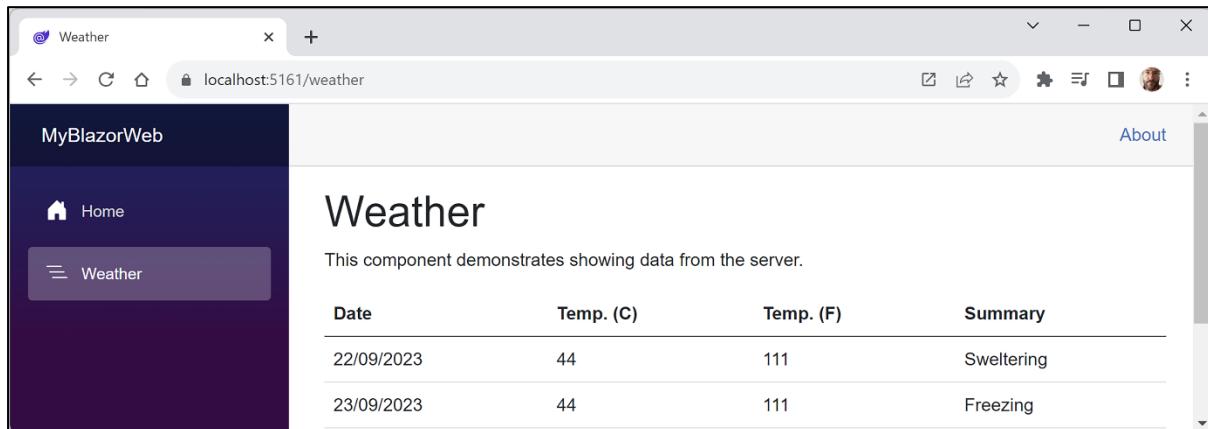
- Solution 'PracticalApps' (9 of 9 projects)
 - Northwind.DataContext.Sqlite
 - Northwind.DataContext.SqlServer
 - Northwind.EntityModels.Sqlite
 - Northwind.EntityModels.SqlServer
 - Northwind.Mvc**
 - Northwind.Razor.Employees
 - Northwind.UnitTests
 - Northwind.Web
 - Northwind.WebApi

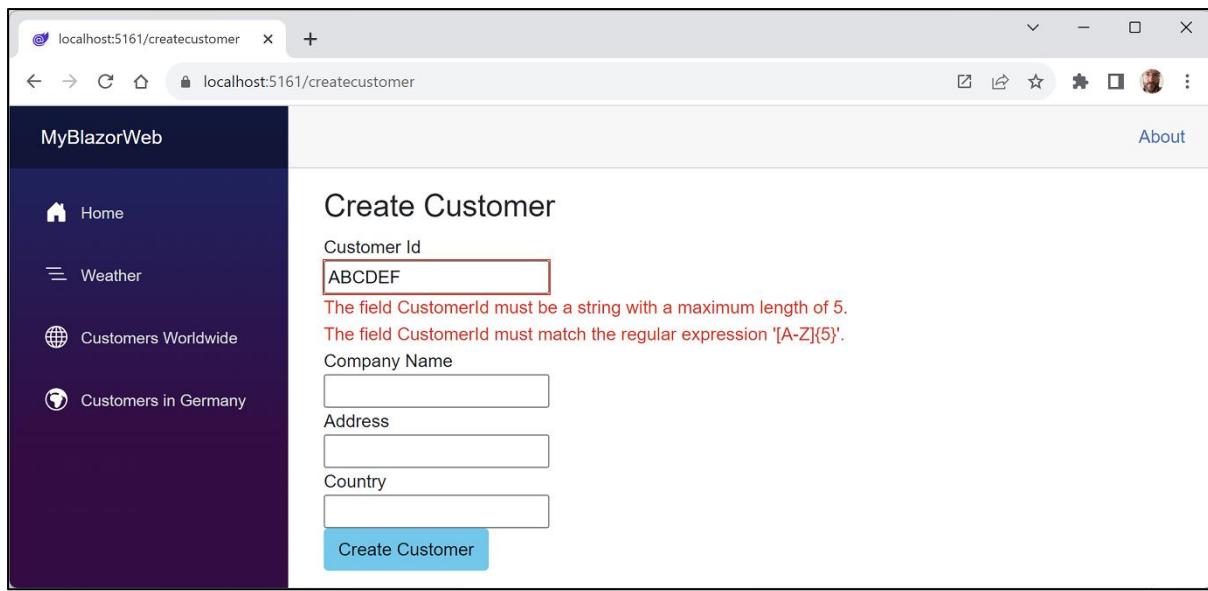
Live Share PREVIEW

The screenshot shows a web browser window with the title "Customers in UK - Northwind.Mvc". The address bar displays "localhost:5141/Home/Customers?country=UK". The page header includes "Northwind.Mvc" and navigation links for "Home" and "Privacy", along with "Register" and "Login" buttons. The main content area is titled "Customers in UK" and contains a table with the following data:

Company Name	Contact Name	Address	Phone
Around the Horn	Thomas Hardy	120 Hanover Sq. London UK WA1 1DP	(171) 555-7788
B's Beverages	Victoria Ashworth	Fauntleroy Circus London UK EC2 5NT	(171) 555-1212
Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery London UK WX1 6LT	(171) 555-2282

Chapter 16: Building User Interfaces Using Blazor





Chapter 17: Epilogue

C# 12 and .NET 8
Modern Cross-Platform Development Fundamentals
Start building websites and services with ASP.NET Core 8, Blazor, and EF Core 8
Eighth Edition
Mark J. Price

- **C# language**, including new C# 12 features, object-oriented programming, debugging, and unit testing.
- **.NET libraries**, including numbers, text, regular expressions, collections, file I/O, and data with EF Core and SQLite.
- **Websites and web services** with ASP.NET Core and Blazor.

Apps and Services with .NET 8
Build practical projects with Blazor, .NET MAUI, gRPC, GraphQL, and other enterprise technologies
Second Edition
Mark J. Price

- **More libraries**: Internationalization, multitasking, and third-party packages.
- **More data**: SQL Server and Cosmos DB.
- **More services**: Minimal APIs, caching, queuing, GraphQL, gRPC, SignalR, and Azure Functions.
- **More user interfaces**: ASP.NET Core MVC, Blazor, and .NET MAUI.

Tools and Skills for .NET 8 Pros
Learn professional skills to design, debug, test, and deploy your solutions and get the career you want
First Edition
Mark J. Price

- **Tools**: IDEs, debugging, memory analysis, and AI assistants.
- **Testing**: Unit, integration, performance, system, and web, including DI and IoC.
- **Design**: Patterns and architecture.
- **Deploy**: Continuous Integration/Deployment, and Azure hosting.
- **Career**: Interview preparation.

