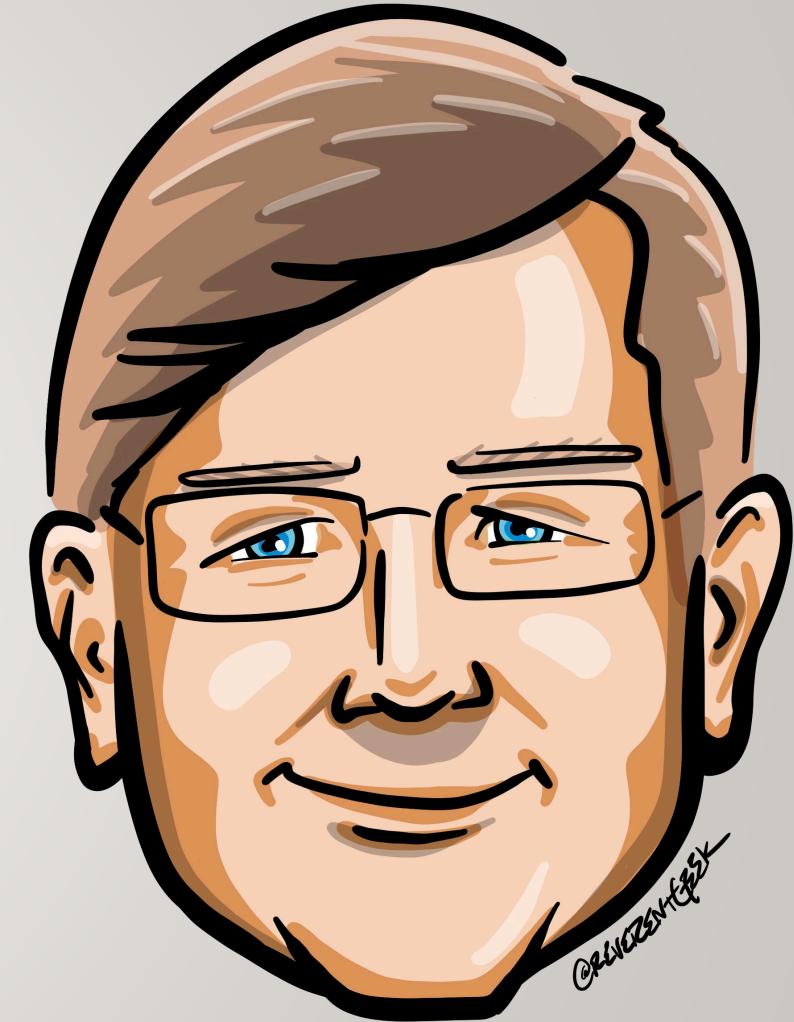


A photograph of a stack of books and an open laptop on a wooden surface. In the foreground, an open book lies flat, its pages fanned out. Behind it, a laptop is open, its screen facing right. To the right of the laptop, there's a stack of several closed books. The background is blurred, showing more books on shelves, creating a library-like atmosphere.

Building Great Libraries

Who is Chad Green

- ✉ chadgreen@chadgreen.com
- 💬 TaleLearnCode
- 🌐 ChadGreen.com
- 🐦 ChadGreen & TaleLearnCode
- linkedin ChadwickEGreen



Public Service Announcement



Listen to your body





Listen to your body

Preamble

Building Great Libraries

What is a class library?

collection of precompiled code
modules, functions, classes, and
resources





What is a class library?

designed to be reused and shared



What is a class library?

ready-made components



What is a class library?

leverage to expedite development process



What is a class library?

organized into namespaces



What is a class library?

promote code reuse and modular development

What is a class library?

provided by the programming languages or created and distributed by third-parties



What is a class library?

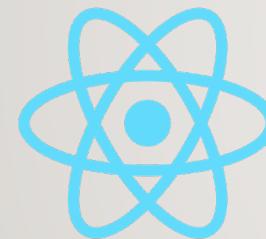
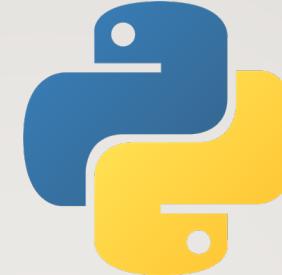
Reusable collection of prewritten code modules providing common functionalities to developers, enabling them to build software more efficiently and with less effort.



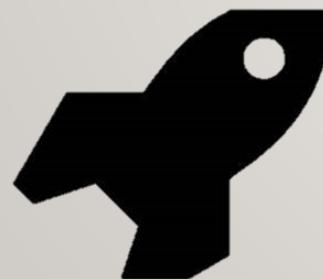
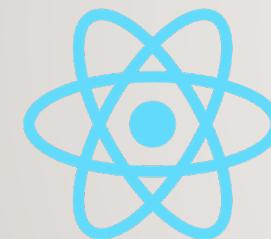
Examples of Class Libraries



Examples of Class Libraries



Examples of Class Libraries





Class Libraries in .NET

.NET Framework Class Library (FCL)



Class Libraries in .NET

.NET Framework Class Library (FCL)

.NET Core Class Library

Class Libraries in .NET

.NET Framework Class Library (FCL)

.NET Core Class Library

.NET Standard Library



Class Libraries in .NET

.NET Framework Class Library (FCL)

.NET Core Class Library

.NET Standard Library

Custom Class Libraries



Class Libraries in .NET

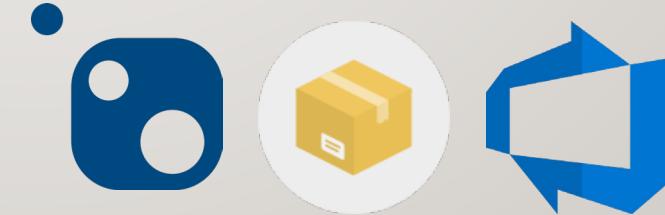
.NET Framework Class Library (FCL)

.NET Core Class Library

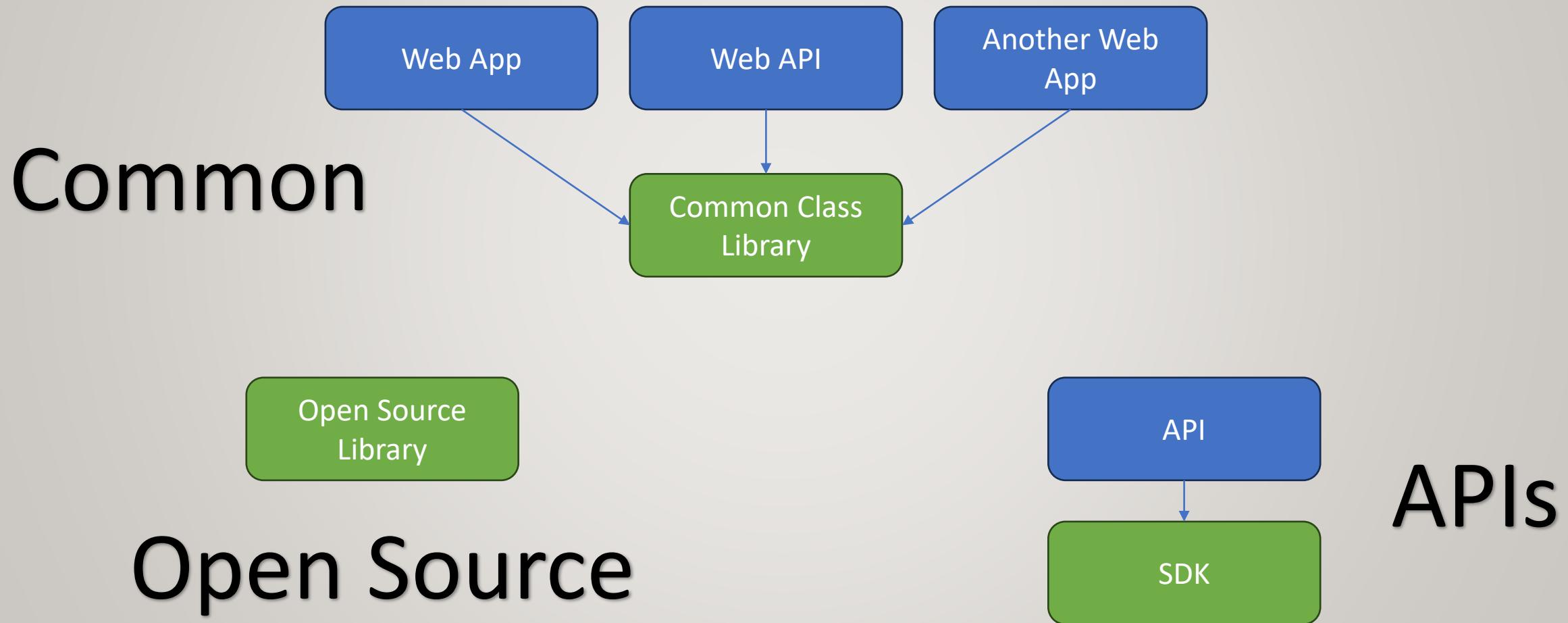
.NET Standard Library

Custom Class Libraries

Packages



What are we talking about?





Best Practices

Building Great Libraries

Best Practices

Define a clear and
focused purpose



Best Practices

Design and reusability
and modularity





Best Practices

Provide clear
documentation

Best Practices

Follow naming
conventions and
guidelines



Best Practices

Implement error
handling and exception
management





Best Practices

Perform thorough
testing



Best Practices

Consider performance
and scalability



Best Practices

Versioning and backward compatibility



Best Practices

Use source control and
package management



Best Practices

Solicit and incorporate
user feedback

Best Practices

- Define a clear and focused purpose
- Design for reusability and modularity
- Provide clear documentation
- Follow naming conventions and guidelines
- Implement error handling and exception management
- Perform thorough testing
- Consider performance and scalability
- Versioning and backward compatibility
- Use source control and package management
- Solicit and incorporate user feedback

Additional Information

Building Great Libraries

Additional Information

Coding Standards

Additional Information

Coding Standards

Common C# Coding Conventions



Additional Information

Coding Standards

Design Guidelines for Developing Class Libraries



Additional Information

Documentation

Project Readme

Documentation Website

Package Readme

XML Documentation

Additional Information

Documentation

Project Readme

Documentation Website

Package Readme

XML Documentation

Additional Information

Documentation

XML Documentation

```
/// <summary>
/// Returns the sum of <paramref name="addend1"/> and <paramref name="addend2"/>.
/// </summary>
/// <param name="addend1">The first number to be added.</param>
/// <param name="addend2">The second number to be added.</param>
/// <returns>The sum of <paramref name="addend1"/> and <paramref name="addend2"/>.</returns>
1 reference | Chad Green, 1 day ago | 1 author, 1 change
public double Add(double num1, double num2)
{
    return num1 + num2;
}
```

Additional Information



Documentation

XML Documentation

- General tags
 - summary
 - remarks
- Document members
 - returns
 - param
 - paramref
 - exception
 - value
- Format document output
 - para
 - list
 - c
 - code
 - example
- Reuse documentation text
 - inheritdoc
 - include
- Generate links and references
 - see
 - seealso
- Generic types and methods
 - typeparam
- User-defined tags

Additional Information

Versioning

Semantic Versioning

7.20.2023

Major.Minor.Patch



Additional Information

Publishing

7.20.2023-preview

1.0.123-dev



Additional Information

Breaking Changes



Additional Information

Breaking Changes

Virtual Methods



Additional Information

Breaking Changes

Method Signatures



Additional Information

Breaking Changes

Obsolete Attribute



Additional Information

Breaking Changes

Optional Arguments

Coding Demonstration

Building Great Libraries

Coding Demonstration

```
1 1 namespace TaleLearnCode.MathOperationsLibrary
2 2 {
3 3     1 reference | Chad Green, 19 minutes ago | 1 author, 1 change
4 4     public interface IMathOperations
5 5     {
6 6         /// <summary>
7 7         /// Returns the sum of <paramref name="addend1"/> and <paramref name="addend2"/>.
8 8         /// </summary>
9 9         /// <param name="addend1">The first number to be added.</param>
10 10        /// <param name="addend2">The second number to be added.</param>
11 11        /// <returns>The sum of <paramref name="addend1"/> and <paramref name="addend2"/>.</returns>
12 12        1 reference | Chad Green, 19 minutes ago | 1 author, 1 change
13 13        double Add(double addend1, double addend2);
14 14
15 15        /// <summary>
16 16        /// Returns the results of removing <paramref name="subtrahend"/> from the <paramref name="minuend"/>.
17 17        /// </summary>
18 18        /// <param name="minuend">: a number from which the subtrahend is to be subtracted</param>
19 19        /// <param name="subtrahend">The number that is to be subtracted from a minuend.</param>
20 20        /// <returns>The results of removing the <paramref name="subtrahend"/> from the <paramref name="minuend"/>.</returns>
21 21        1 reference | Chad Green, 19 minutes ago | 1 author, 1 change
22 22        double Subtract(double minuend, double subtrahend);
23 23
24 24        /// <summary>
25 25        /// Returns the product of multiplying the <paramref name="multiplicand"/> by the <paramref name="multiplier"/>.
26 26        /// </summary>
27 27        /// <param name="multiplicand">The number that is to be multiplied by the <paramref name="multiplier"/>.</param>
28 28        /// <param name="multiplier">The number by which the <paramref name="multiplicand"/> is multiplied</param>
29 29        /// <returns>The product of multiplying the <paramref name="multiplicand"/> by the <paramref name="multiplier"/>.</returns>
30 30        1 reference | Chad Green, 19 minutes ago | 1 author, 1 change
31 31        double Multiply(double multiplicand, double multiplier);
32 32
33 33        /// <summary>
34 34        /// Returns the calculation of the number of times one number (<paramref name="dividend"/>) is contained within another (<paramref name="divisor"/>).
35 35        /// </summary>
36 36        /// <param name="dividend">a number to be divided</param>
37 37        /// <param name="divisor">the number by which a dividend is divided</param>
38 38        /// <returns></returns>
39 39        1 reference | Chad Green, 19 minutes ago | 1 author, 1 change
40 40        double Divide(double dividend, double divisor);
41 41    }
42 42 }
```

Coding Demonstration

```
1  using System;
2
3  namespace TaleLearnCode.MathOperationsLibrary
4  {
5      4 references | Chad Green, 20 minutes ago | 1 author, 1 change
6      public class MathOperationsException : Exception
7      {
8          0 references | Chad Green, 20 minutes ago | 1 author, 1 change
9          public MathOperationsException() { }
10
11         1 reference | Chad Green, 20 minutes ago | 1 author, 1 change
12         public MathOperationsException(string message) : base(message) { }
13
14         0 references | Chad Green, 20 minutes ago | 1 author, 1 change
15         public MathOperationsException(string message, Exception innerException)
16             : base(message, innerException) { }
17     }
18 }
```

Coding Demonstration

```
1  namespace TaleLearnCode.MathOperationsLibrary
2  {
3
4      0 references | Chad Green, 21 minutes ago | 1 author, 1 change
5      public class MathOperations : IMathOperations
6      {
7          /// <summary> Returns the sum of addend1 and addend2.
8          1 reference | Chad Green, 21 minutes ago | 1 author, 1 change
9          public double Add(double num1, double num2)
10         {
11             return num1 + num2;
12         }
13
14
15
16
17
18          /// <summary> Returns the results of removing subtrahend from the minuend.
19          1 reference | Chad Green, 21 minutes ago | 1 author, 1 change
20          public double Subtract(double num1, double num2)
21          {
22              return num1 - num2;
23          }
24
25
26
27
28
29          /// <summary> Returns the product of multiplying the multiplicand by the multipl ...
30          1 reference | Chad Green, 21 minutes ago | 1 author, 1 change
31          public double Multiply(double num1, double num2)
32          {
33              return num1 * num2;
34          }
35
36
37
38
39
40          /// <summary> Returns the calculation of the number of times one number ( divide ...
41          1 reference | Chad Green, 21 minutes ago | 1 author, 1 change
42          public double Divide(double num1, double num2)
43          {
44
45              if (num2 == 0)
46              {
47
48                  //throw new DivideByZeroException("Cannot divide by zero
49                  throw new MathOperationsException("Cannot divide by zero");
50              }
51
52              return num1 / num2;
53
54
55
56
57
58 }
```

Coding Demonstration

```
readme.md – MathOperationsLibrary × README.md – BuildingGreatLibraries ×

Returns the sum of the supplied addends.

### Subtract
Returns the results of removing the subtrahend from the minuend.

### Multiply
Returns the product of multiplying the multiplicand by the multiplier.

### Division
Returns the calculation of the number of times one number (dividend) is contained within another (divisor).
```

TaleLearnCode Math Operations Library

This is a fully operational library providing simple mathematical operations. The intent of this library is to demonstrate how to build great libraries and is a part of the [Build Great Libraries](#) presentation presented by Chad Green.

Operations

Add

Returns the sum of the supplied addends.

Subtract

Returns the results of removing the subtrahend from the minuend.

Multiply

Returns the product of multiplying the multiplicand by the multiplier.

Division

Returns the calculation of the number of times one number (dividend) is contained within another (divisor).



created with the free version of [Markdown Monster](#)

Coding Demonstration

The screenshot shows the Visual Studio Properties window for the `MathOperationsLibrary` project. The `MathOperationsException.cs`, `MathOperations.cs`, and `IMathOperations.cs` files are listed in the tabs at the top. The left sidebar has sections for Application, Build, Package (which is expanded), General, License, Symbols, Code Analysis, Debug, and Resources. The General section is selected in the main pane. Under General, there is a checkbox for "Generate NuGet package on build" which is checked. Below it, the "Package ID" field contains `$(AssemblyName)`, with the value `Glennis.PPM.ListingServices` shown below. The "Title" field contains `Math Operations Library (Demo)`. The "Package Version" field contains `0.0.1-beta`. The "Authors" field contains `TaleLearnCode`. The "Company" field contains `$(Authors)`, with the value `TaleLearnCode` shown below. The "Product" field contains `Building Great Libraries`.

Coding Demonstration

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netstandard2.0</TargetFramework>
5     <RootNamespace>TaleLearnCode.MathOperationsLibrary</RootNamespace>
6     <AssemblyName>Glennis.PPM.ListingServices</AssemblyName>
7     <Version>0.0.1-beta</Version>
8     <Authors>TaleLearnCode</Authors>
9     <Product>Building Great Libraries</Product>
10    <Description>Provides a set of basic mathematic operation methods. This is for presentation purposes only.</Description>
11    <Copyright>© 2023 Green Events and Technology. All rights reserved.</Copyright>
12    <RepositoryUrl>https://github.com/TaleLearnCode/BuildingGreatLibraries</RepositoryUrl>
13    <PackageProjectUrl>https://www.chadgreen.com/presentations/building-great-libraries</PackageProjectUrl>
14    <Title>Math Operations Library (Demo)</Title>
15    <RepositoryType>git</RepositoryType>
16    <PackageTags>math;demo;presentation</PackageTags>
17    <PackageReleaseNotes>Producing the initial demo release of the TaleLearnCode Math Operations Library.</PackageReleaseNotes>
18    <NeutralLanguage>en-US</NeutralLanguage>
19    <GeneratePackageOnBuild>True</GeneratePackageOnBuild>
20    <PackageReadmeFile>readme.md</PackageReadmeFile>
21    <PackageLicenseExpression>MIT</PackageLicenseExpression>
22  </PropertyGroup>
23
24  <ItemGroup>
25    <None Update="readme.md">
26      <Pack>True</Pack>
27      <PackagePath>\</PackagePath>
28      <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
29    </None>
30  </ItemGroup>
31
32</Project>
```

Thank You

Slides & Demo Code

✉ chadgreen@chadgreen.com

💬 TaleLearnCode

🌐 ChadGreen.com

🐦 ChadGreen & TaleLearnCode

linkedin ChadwickEGreen

