CSC445 – Modern Programming Languages

Lecture 2

The C# Language

Introducing the C#

- C# or CSharp
- Created by Microsoft
- Object Oriented Programming
- Can be used to build the following applications:
    - Mobile applications
    - Desktop applications
    - Web applications
    - Web services
    - Web sites
    - Games
    - VR
    - Database applications
    - And much, much more!

Why Use C#?

- It’s one of the most popular programming language in the world.
- Easy to learn and simple to use.
- It has a huge community support.
- C# is an OOP, Code reuse, Lowering the cost.
- C, C++, Java programmers can easily switch.
- It’s a modern programming language.

Language Versions

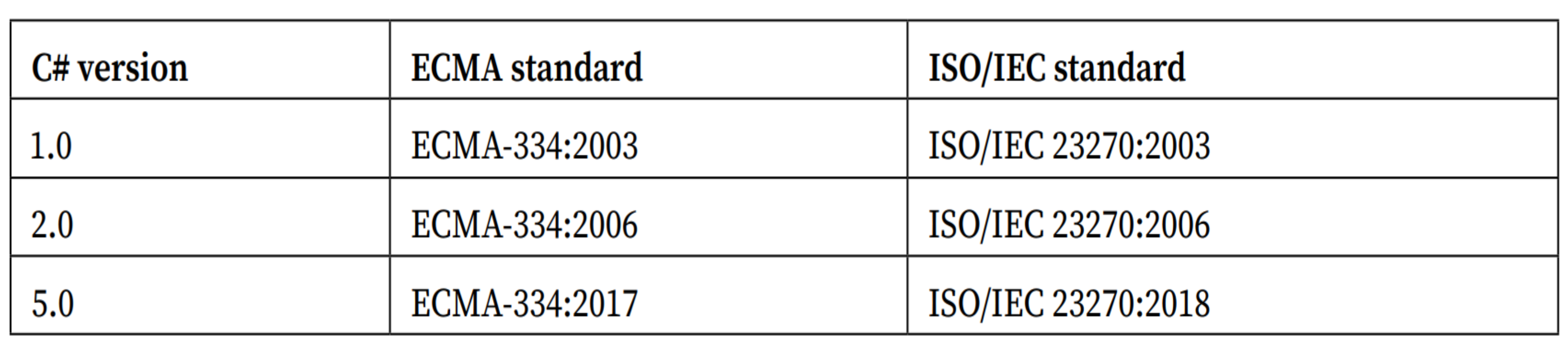
- COOL (C like objected oriented language)
- C# 1 February 2002
- C# 1.2 Released with VS 2003
- C# 2 in 2005 – Generics …
- C# 3 in 2007 – LINQ …
- C# 4 in 2010 – Dynamic Types …
- C# 5 in 2012 – Asynchronous Tasks …
- C# 6 in 2015 – Interpolated Strings …
- C# 7 in 2017 – Pattern Matching …

Language Versions

- C# 7.1 in 2017 – Minor refinements
- C# 7.2 in 2017 – More refinements
- C# 7.3 in 2018 – Performance enhancements
- C# 8 in 2019 – Nullable References
- C# 9 in 2020 – Records
- C# 10 in 2021 – Global Namespaces
- C# 11 in 2022 – Required Properties

Understanding Standards

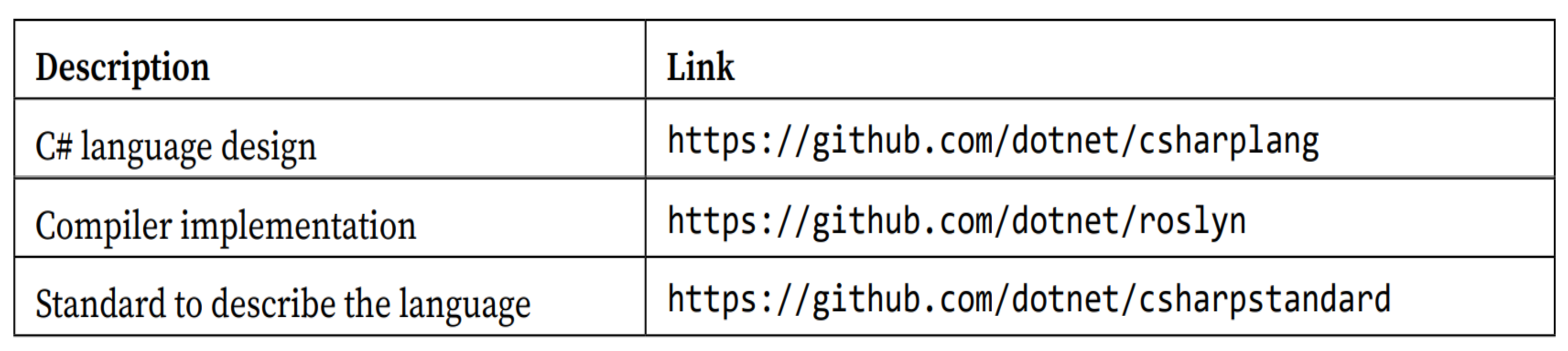
- Microsoft has submitted few versions of C# to standardbred bodies.



C# version	ECMA standard	ISO/IEC standard
1.0	ECMA-334:2003	ISO/IEC 23270:2003
2.0	ECMA-334:2006	ISO/IEC 23270:2006
5.0	ECMA-334:2017	ISO/IEC 23270:2018

Understanding Standards

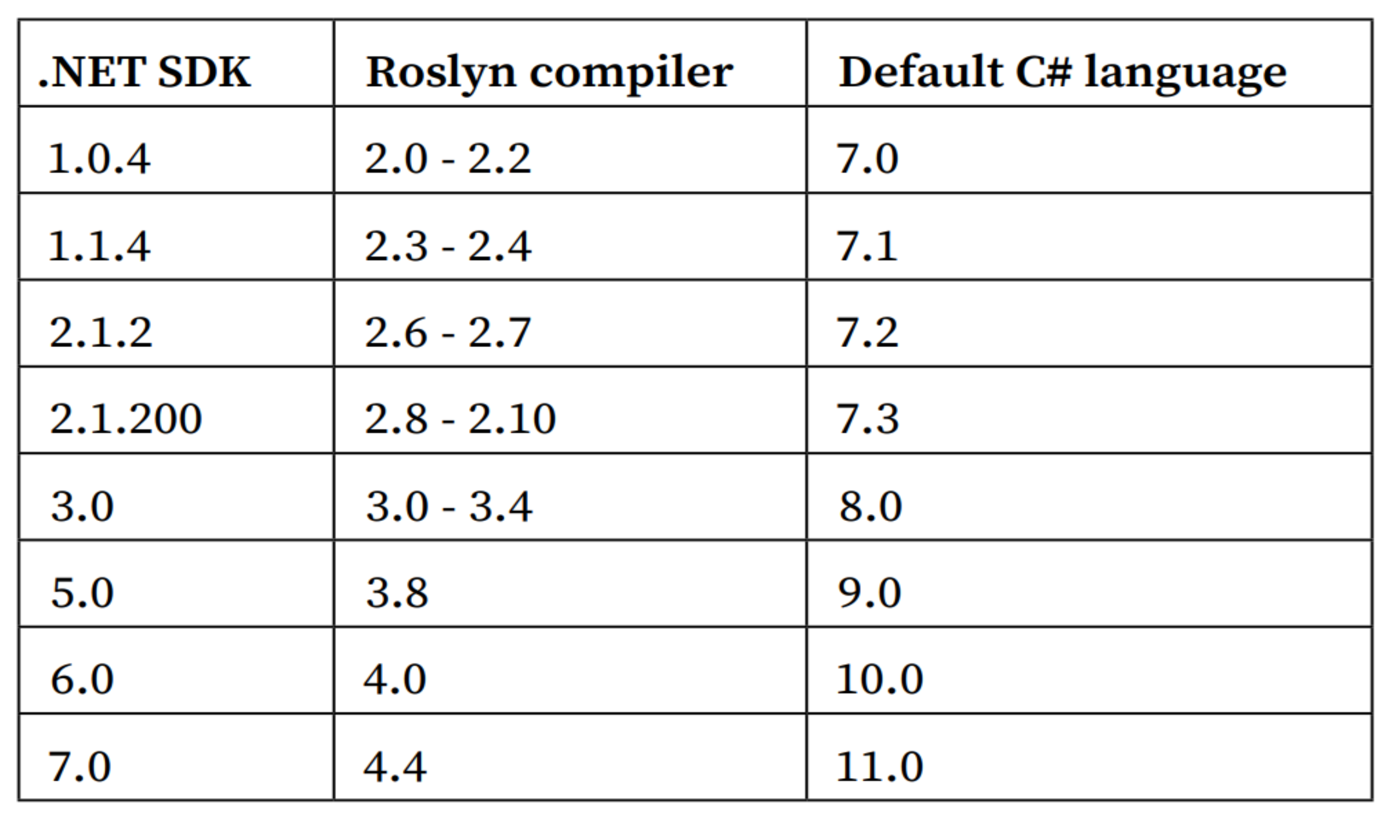
- More practically useful than the ECMA standards are the public GitHub repositories for making the work on C# and related technologies as open as possible.



Description	Link
C# language design	https://github.com/dotnet/csharplang
Compiler implementation	https://github.com/dotnet/roslyn
Standard to describe the language	https://github.com/dotnet/csharpstandard

Discovering Compiler Versions

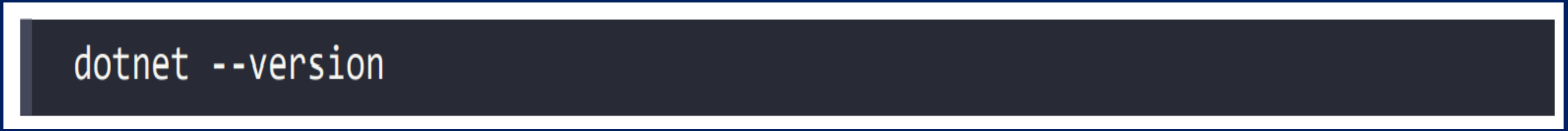
- .NET Compiler for C#, Roslyn, distributed as part of .NET SDK.



.NET SDK	Roslyn compiler	Default C# language
1.0.4	2.0 - 2.2	7.0
1.1.4	2.3 - 2.4	7.1
2.1.2	2.6 - 2.7	7.2
2.1.200	2.8 - 2.10	7.3
3.0	3.0 - 3.4	8.0
5.0	3.8	9.0
6.0	4.0	10.0
7.0	4.4	11.0

How to output SDK version?

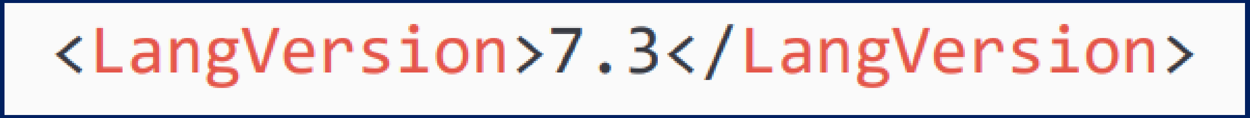
- Go to Windows terminal or Command Prompt on windows platform
- On macOS, start terminal
- Determine which version of .NET SDK, enter the following command:



```
dotnet --version
```

Enabling a specific compiler version

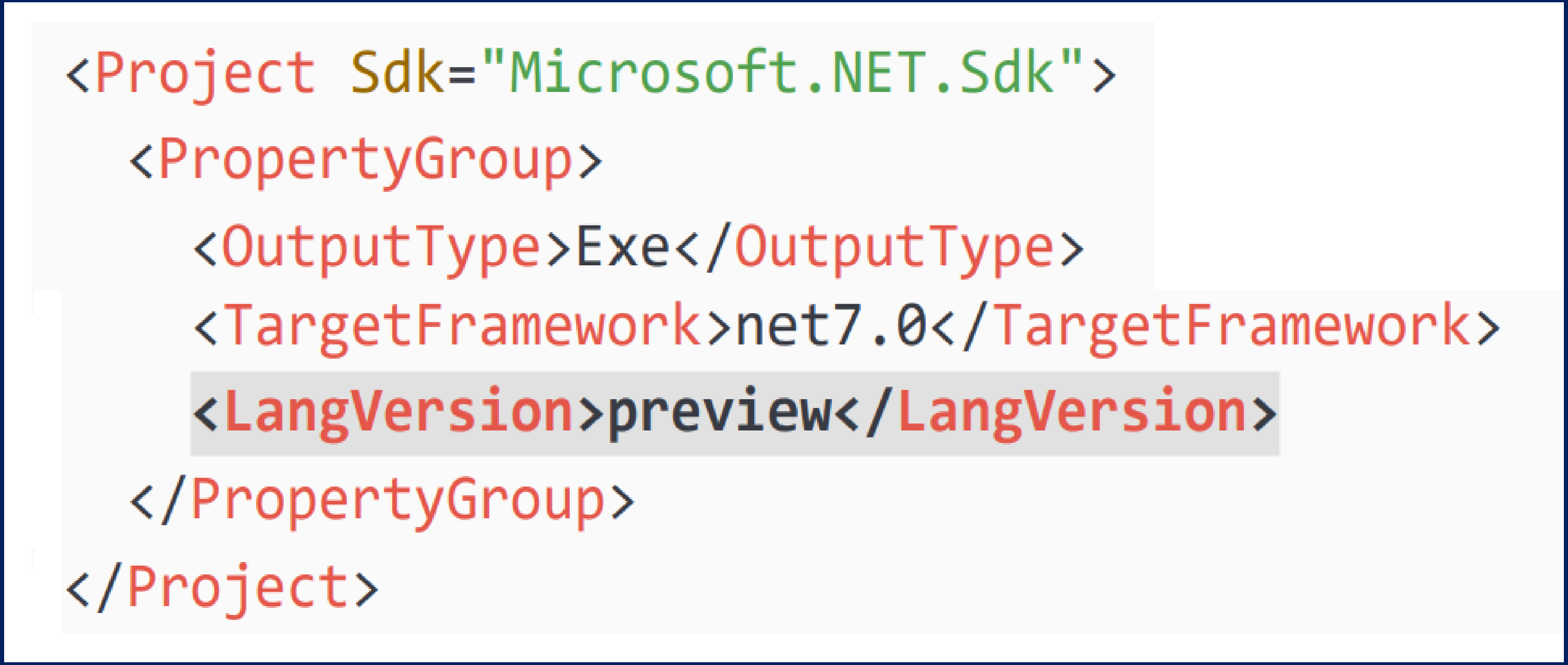
- Developer tools like Visual Studio and the dotnet command-line interface assume that you want to use the latest major version of a C# language compiler by default.
- To use the improvements in a C# point release like 7.1, 7.2, or 7.3, you had to add a configuration element to the project file, as shown in the following markup:



```
<LangVersion>7.3</LangVersion>
```

Change version by editing project file

- After creating a new project, you can edit the .csproj file and add the element, as shown highlighted in the following markup:



```xml
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net7.0</TargetFramework>
    <LangVersion>preview</LangVersion>
  </PropertyGroup>
</Project>
```

C# Grammer & Vocabulary

- To learn simple C# language features, you can use .NET Interactive Notebooks, which remove the need to create an application of any kind.
- To learn some other C# language features, you will need to create an application. The simplest type of application is a console app.

Understanding C# Grammer

- Grammer includes statements and blocks.
- Order matters.
  - “The black cat” English adjective before cat
  - “le chat noir” French adjective after cat
- Statements, Variable and Expressions and Comments