

# Programming 3 Advanced

## C# and .Net, C# basics

Tomasz Herman

Faculty of Mathematics and Information Science  
Warsaw University of Technology

Lecture 1, 7 października 2024



- 1 .NET Overview
- 2 C# Overview
- 3 The history of C#
- 4 C# First Program



- Common Language Runtime (CLR)
  - Common Type System (CTS)
  - JIT compiler
    - Common Intermediate Language (CIL) → native machine code
  - Garbage Collector (GC)
  - Supported Languages: C# , F# , Visual Basic, and many others
- Base Class Library (BCL)



### Web applications

- ASP.NET

### Desktop applications

- Windows Forms
- Windows Presentation Foundation (WPF)
- WinUI 3

### Mobile and Desktop cross-platform

- MAUI

### Games

- Unity



- dotnet command
- msbuild
- NuGet



- .NET Framework
- Mono
- .NET (Core)



# Key features of C#

- Object-Oriented
- Type safety
  - strongly, statically typed
- Rich Standard Library
- Language Integrated Query (LINQ)
- Garbage Collection
- Asynchronous Programming
- Cross-Platform Development



# C++ similarities

- C derived syntax
- Object-Oriented
- Type safety
- Exception handling
- operator and method overloading
- namespaces
- templates (C++ )/generics (C# )
- default parameters
- preprocessor directives





- Value and Reference Types
- Operators
- Object-Oriented Components: Classes, Structs, Interfaces, Properties
- Functional Components: Delegates, Lambdas, Events
- Generics
- Unsafe Code



# C# version 1.0

Released January 2002

## Major features

- Classes
- Structs
- Interfaces
- Events
- Properties
- Delegates
- Operators and expressions
- Statements
- Attributes

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-10-1>

# C# version 2.0

Released November 2005

## New features

- **Generics**
- Partial types
- **Anonymous methods**
- **Nullable value types**
- **Iterators**
- Covariance and contravariance

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-20>



# C# version 3.0

Released November 2007

## New features

- **Auto-implemented properties**
- Anonymous types
- Query expressions
- **Lambda expressions**
- Expression trees
- **Extension methods**
- **Implicitly typed local variables**
- Partial methods
- **Object and collection initializers**

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-30>

# C# version 4.0

Released April 2010

## New features

- **Dynamic binding**
- **Named/optional arguments**
- Generic covariant and contravariant
- Embedded interop types

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-40>



# C# version 5.0

Released August 2012

## New features

- **Asynchronous members**
- Caller info attributes

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-50>



# C# version 6.0

Released July 2015

## New features

- Static imports
- Exception filters
- Auto-property initializers
- Expression bodied members
- Null propagator
- **String interpolation**
- nameof operator
- Index initializers
- Await in catch/finally blocks
- Default values for getter-only properties

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-60>

# C# version 7.0

Released March 2017

## New features

- **Out variables**
- **Tuples and deconstruction**
- **Pattern matching**
- **Local functions**
- Expanded expression bodied members
- Ref locals
- Ref returns
- Discards
- Binary Literals and Digit Separators
- Throw expressions

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-70>



# C# version 7.1, 7.2, 7.3

Released August 2017, November 2017, May 2018

## New features

- In variables
- readonly structs
- async Main method
- Many other small changes



# C# version 8.0

Released September 2019

## New features

- Readonly members
- Default interface methods
- **Pattern matching enhancements**
- **Using declarations**
- Static local functions
- Disposable ref structs
- **Nullable reference types**
- Asynchronous streams
- Indices and ranges
- Null-coalescing assignment

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-80>

# C# version 9

Released November 2020

## New features

- **Records**
- Init only setters
- **Top-level statements**
- Pattern matching enhancements
- Native sized integers
- Function pointers
- Suppress emitting localsinit flag
- Module initializers
- New features for partial methods
- and other changes

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history#c-version-9>

# C# version 10

Released November 2021

## New features

- **Record structs**
- Improvements of structure types
- Interpolated string handler
- **Global using directives**
- **File-scoped namespace declarations**
- **Lambda expression improvements**
- and other changes

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-10>

# C# version 11

Released November 2022

## New features

- Generic attributes
- **Generic math support**
- Numeric IntPtr and UIntPtr
- List patterns
- **Raw string literals**
- Auto-default struct
- Required members
- ref fields and ref scoped variables
- File local types
- and other changes

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-11>

# C# version 12

Released November 2023

## New features

- **Primary constructors**
- **Collection expressions**
- Inline arrays
- Optional parameters in lambda expressions
- ref readonly parameters
- Alias any type
- Experimental attribute
- Interceptors (*preview*)

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-12>

## New features

- **params collections**
- New lock object
- New escape sequence
- Method group natural type
- Implicit index access
- ref and unsafe in iterators and async methods
- allows ref struct
- ref struct interfaces
- More partial members
- Overload resolution priority

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-13>

# First Program

```
1  using System;
2  public class Program
3  {
4      private static void Main(string[] args)
5      {
6          Console.WriteLine("Hello, World!");
7
8          string name = "Alice";
9          Console.WriteLine($"Hello, {name}!");
10
11         for (int i = 0; i < 5; i++)
12         {
13             Console.WriteLine($"Count: {i}");
14         }
15     }
16 }
```





# Top-level statements

C# 9

```
1 using System;
2
3 Console.WriteLine("Hello, World!");
4
5 string name = "Alice";
6 Console.WriteLine($"Hello, {name}!");
7
8 for (int i = 0; i < 5; i++)
9 {
10     Console.WriteLine($"Count: {i}");
11 }
```

## Documentation

<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/program-structure/top-level-statements>