.NET Introduction

Lesson Objectives

- .NET Introduction
- C# Programming Language Introduction
- Create first program in C# with Visual Studio
  - ✓ Structure of C# program
  - ✓ Basic IO in C# program

Section 1

.NET INTRODUCTION

.NET Introduction

- The .NET Framework is a programming model that supports building and running of software applications for
  - ✓ Windows,
  - ✓ Windows Server,
  - ✓ Windows Phone,
  - ✓ Microsoft Azure,
  - ✓ XML Web services.

.NET Introduction

- Developed by Microsoft and runs on Microsoft Windows.
- The .NET Framework consists of a large class library known as Framework Class Library (FCL), and Common Language Runtime (CLR) which provides language interoperability across several programming languages
  - ✓ *FCL: Framework Class Library*
  - ✓ *CLR: Common Language Runtime*

.NET Introduction

- Written to overcome many of the problems of application development including long development times, inability to change applications quickly, high total cost of ownership of software and ease of deployment.

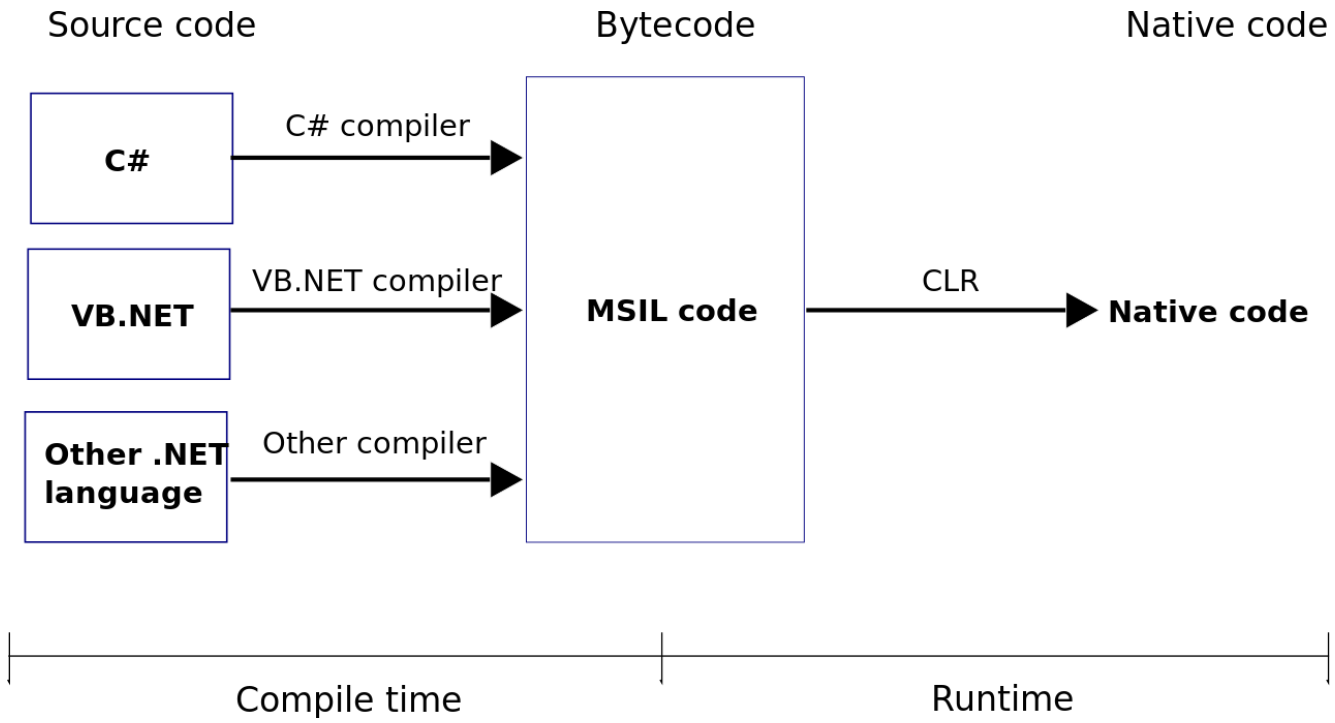
Common Language Runtime

- Manages memory, execution of threads and code, verification of code safety, compilation, and other system services
- Implements code access security
- Ensures code robustness through a strict type-and-code-verification infrastructure called the common type system (CTS).

Common Language Runtime

- Just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing and memory manager removes possibilities of fragmented memory and increases memory locality-of-reference which enhances performance.
- Runtime can be hosted by server-side applications, such as Microsoft SQL Server and Internet Information Services (IIS) that are high on performance

Common Language Runtime



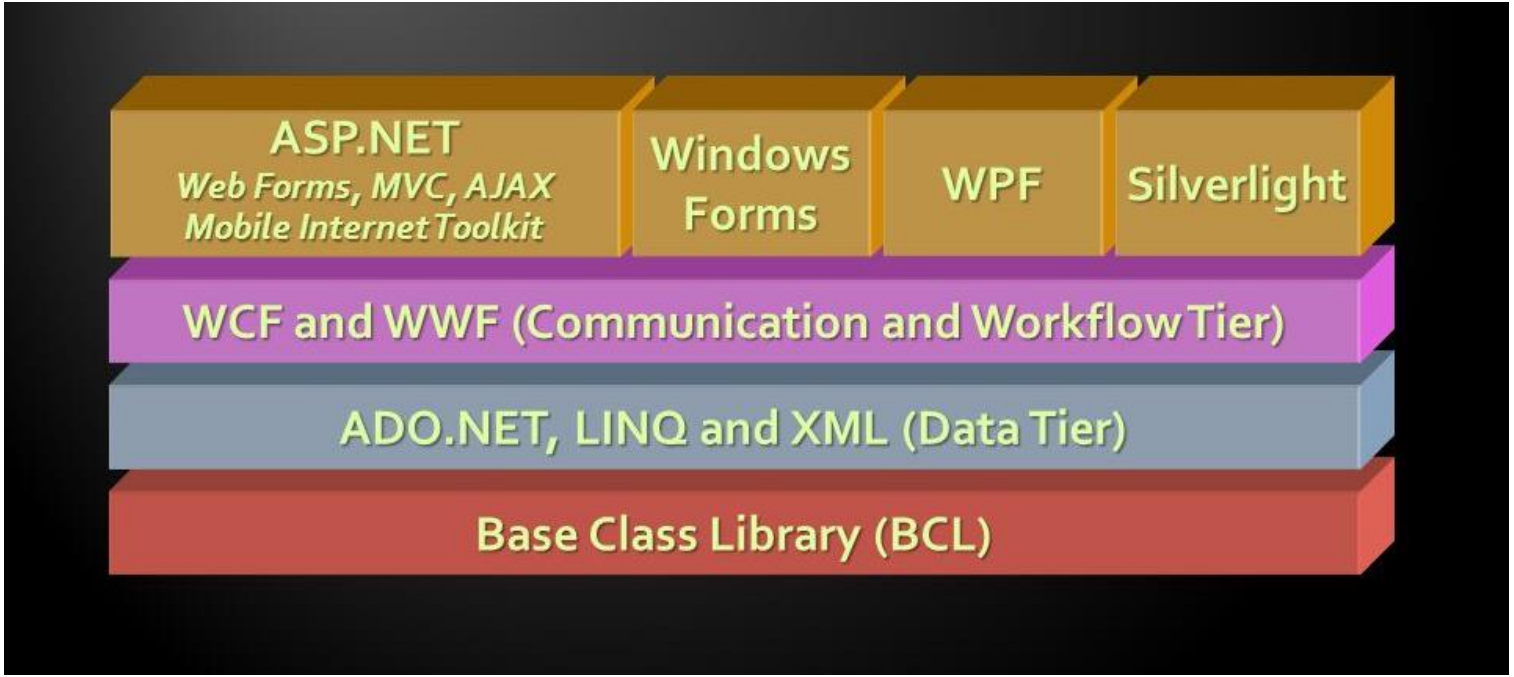
Framework Class Library

- Includes a set of standard class libraries. These class libraries execute common functions, such as graphic rendering, database interaction, and XML document manipulation, among others.

Framework Class Library

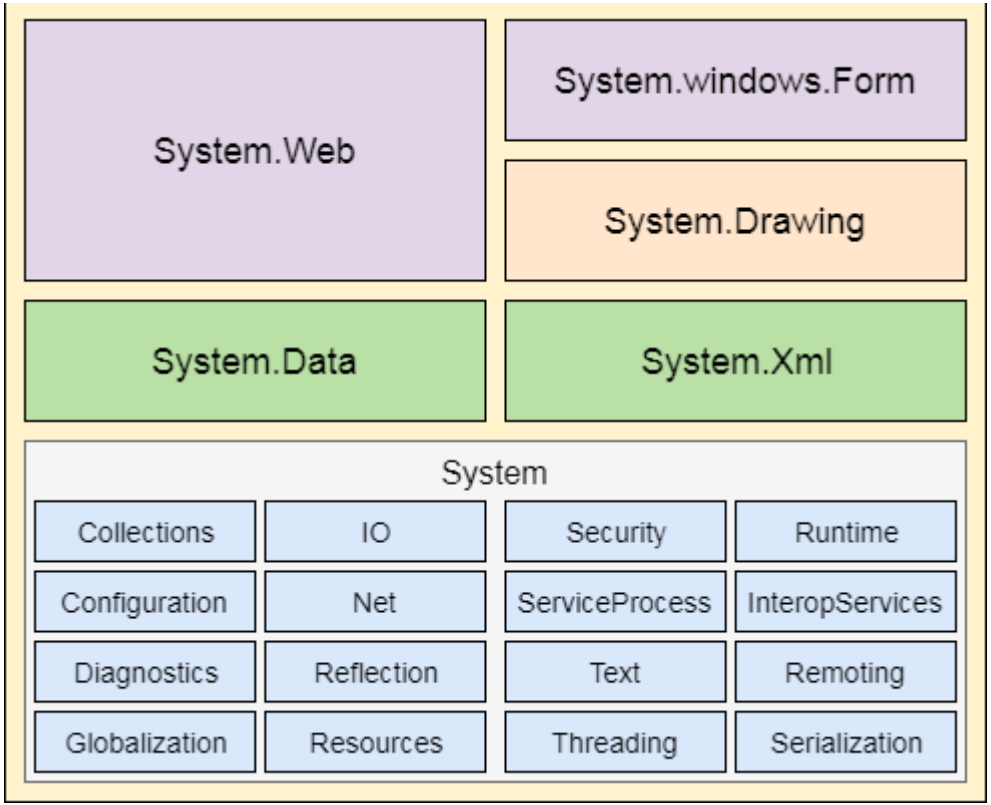
- Collection of reusable types that are closely integrated with the CLR.
- Object-oriented class library which aims to accomplish a range of common programming tasks, such as string management, data collection, database connectivity, and file access

Framework Class Library



Base Class Library

- Is the sub part of the Framework that provides library support to CLR to work properly.
- It includes the System namespace and core types of the .NET framework.



Advantages

1. Less Coding and Increased Reuse of Code
2. Easy Deployment
3. Reliability
4. Security
5. Use across Platforms and Languages
6. Use for Service-Oriented Architecture
7. Integration with Legacy Systems

Types of applications

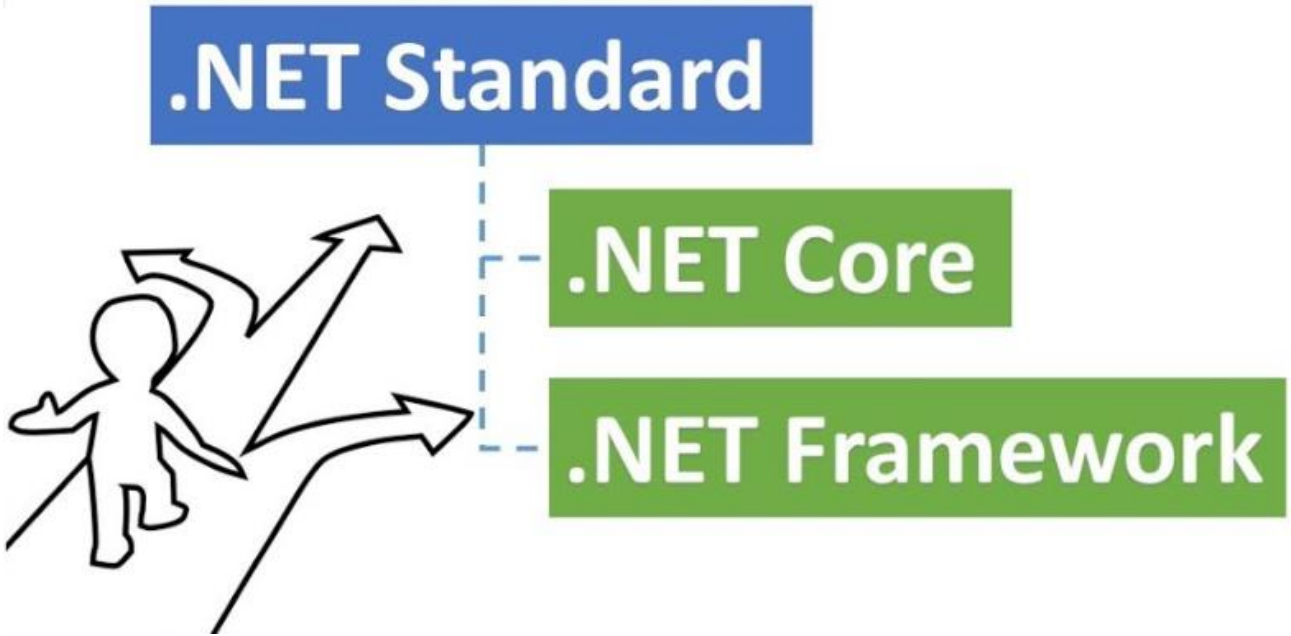
1. Desktop applications
2. Web based applications
3. Applications on embedded systems
4. Others

.NET Framework version history

Version	CLR	Release	Visual Studio
1	1	Jan-02	Visual Studio .NET (2002)
2	2	Oct-05	Visual Studio 2005
3	2	Nov-06	Visual Studio 2005
3.5	2	Nov-07	Visual Studio 2008
4	4	Apr-10	Visual Studio 2010
4.5	4	Aug-12	Visual Studio 2012
4.6	4	Jul-15	Visual Studio 2015
4.7	4	Apr-17	Visual Studio 2017 15.1
4.7.1	4	Oct-17	Visual Studio 2017 15.5
4.7.2	4	Apr-18	Visual Studio 2017 15.8
4.8	4	Apr-19	Visual Studio 2019 16.3

.NET Core

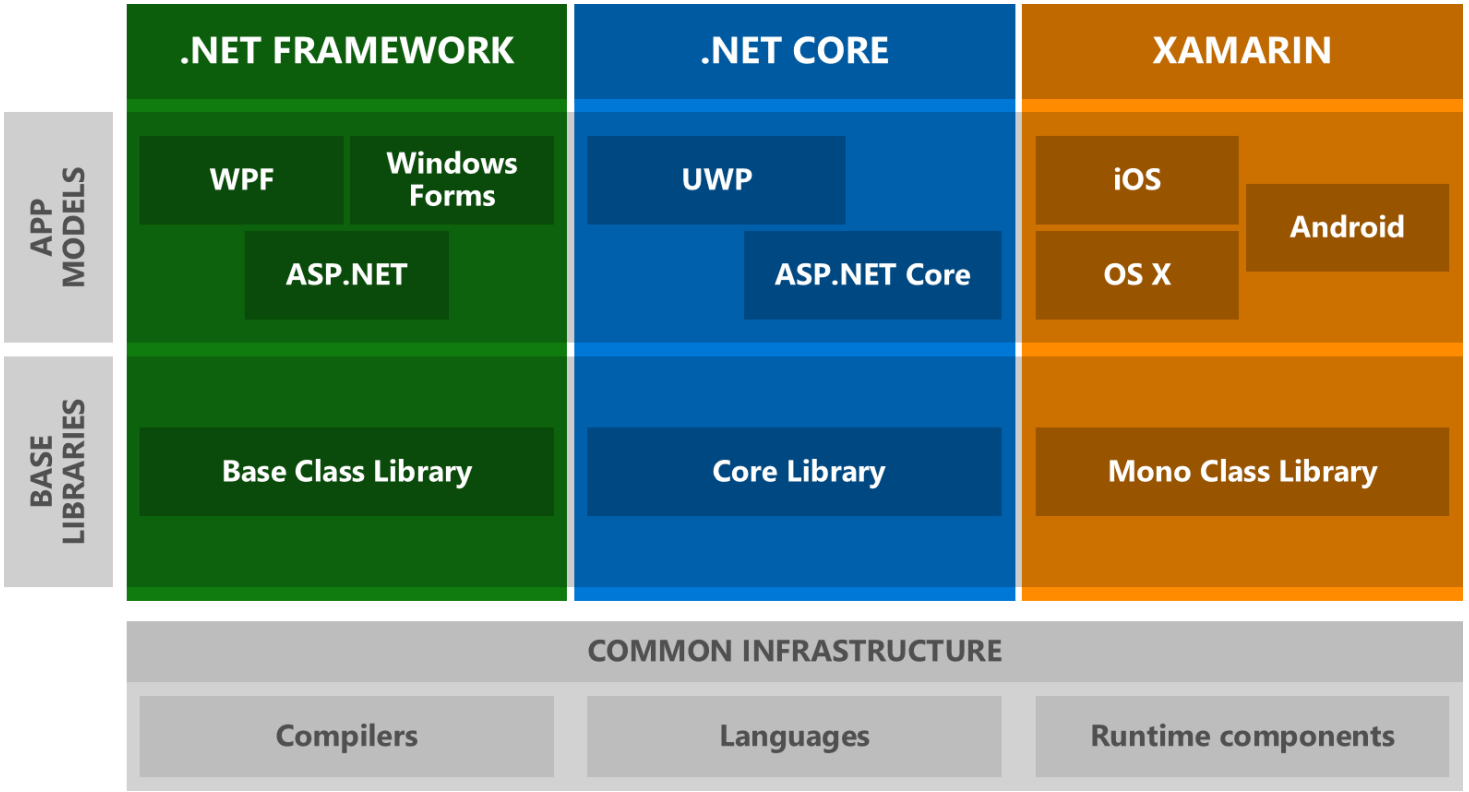
- .NET Core is **NOT** new version of .NET Framework



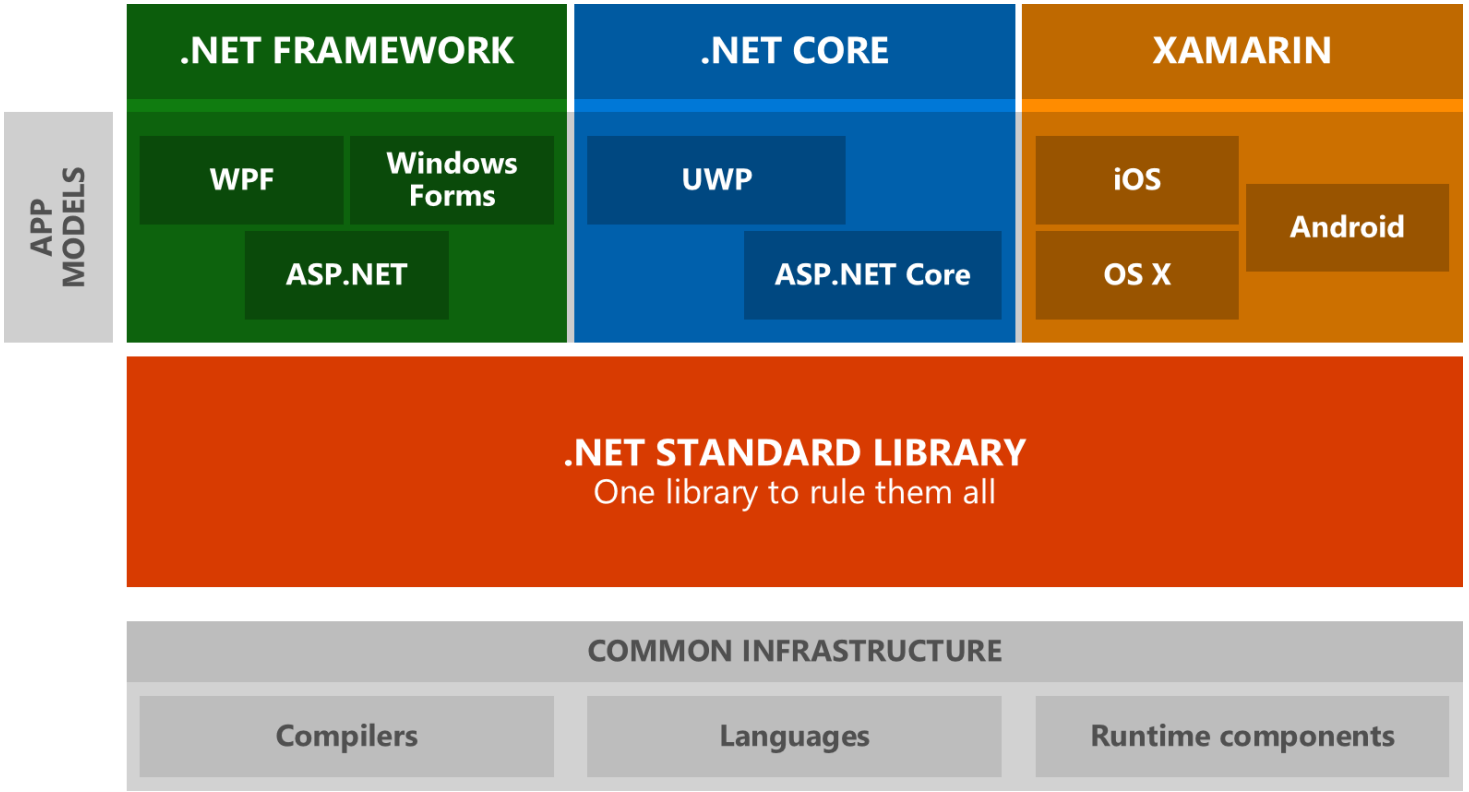
.NET Core

- Create **cross platforms application**.
- Can use Microservices
- Deploy an application to Dockers container.
- Create highly scalable and performable system.
- Create ASP.Net Core, Razor page, UMP, Mobile native app and Blazor application.

.NET Core vs .NET Framework



.NET Standard



Name: **.NET 5**

Date of Birth: **Nov 2020**

Family: **Microsoft**

- Based on:
  - ✓ .NET Framework
  - ✓ .NET Core
  - ✓ .NET Standard
- Strength
  - ✓ Fast Start-up
  - ✓ Low footprint
  - ✓ Lower memory usage

Section 2

C# PROGRAMMING LANGUAGE INTRODUCTION

Using .NET Framework

- A programmer can develop applications using one of the languages supported by .NET.
  - ✓ C#
  - ✓ VB
- These applications make use of the base class libraries provided by the .NET Framework.

C# programming language

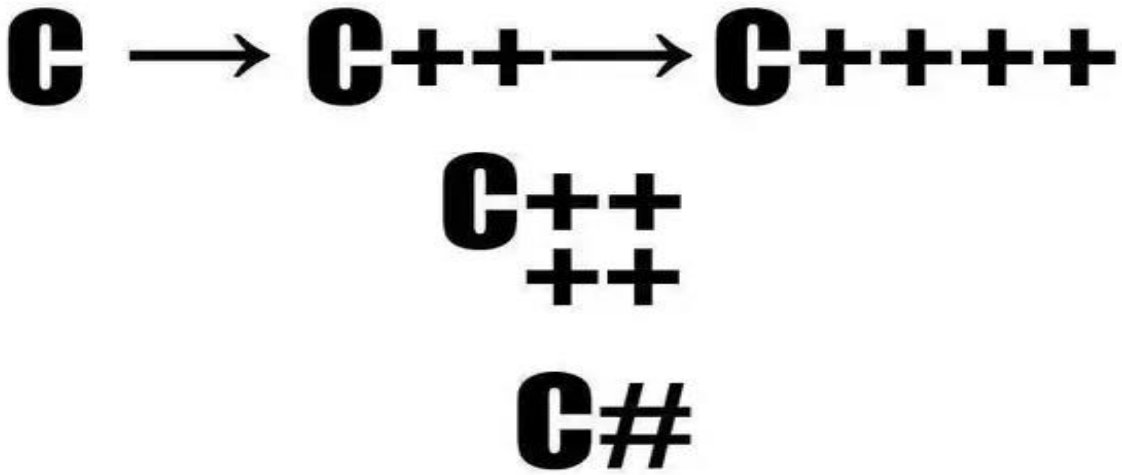
- C# was developed to:
  - ✓ Create a very simple and yet powerful tool for building interoperable, scalable, and robust applications.
  - ✓ Create a complete object-oriented architecture.
  - ✓ Support powerful component-oriented development.
  - ✓ Allow access to many features previously available only in C++ while retaining the ease-of-use of a rapid application development tool such as Visual Basic.
  - ✓ Provide familiarity to programmers coming from C or C++ background.
  - ✓ Allow to write applications that target both desktop and mobile devices.

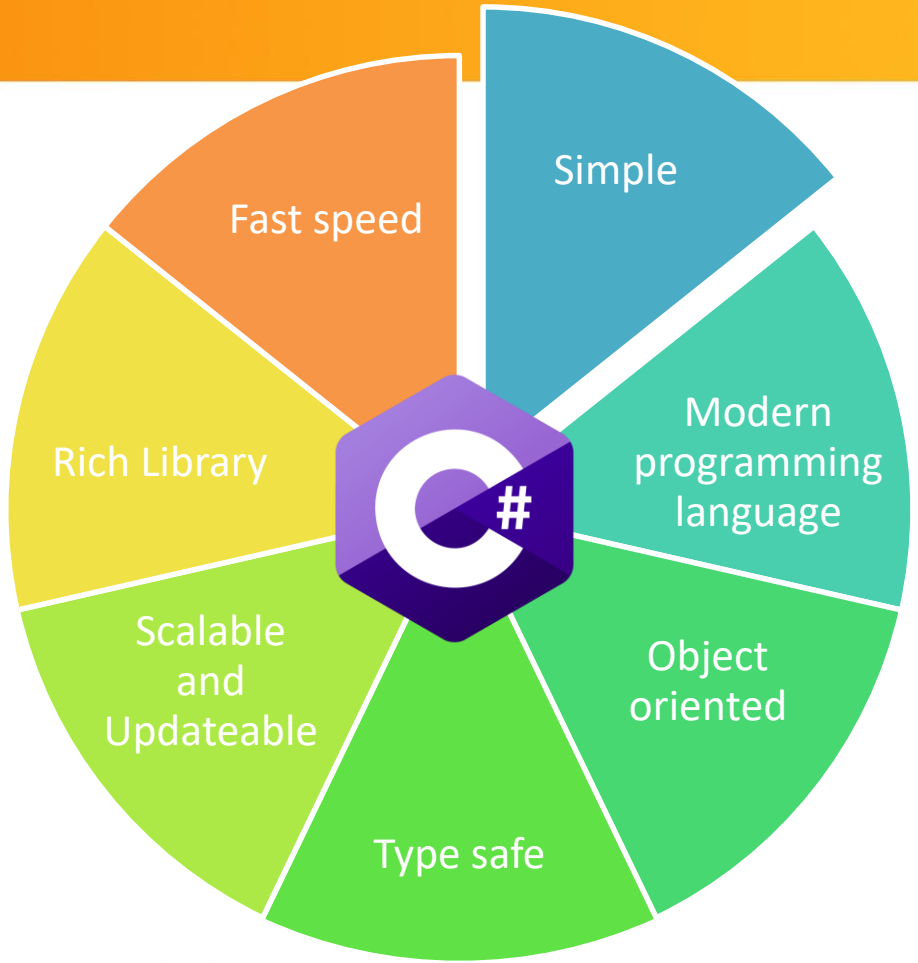
C# programming language

- C# is a programming language designed for building a wide range of applications that run on the .NET Framework.
- C# was developed within .NET framework initiative
- C# was designed by **Anders Hejlsberg**

C#: from code to music

- C# is a programming language designed for building a wide range of applications that run on the .NET Framework.




Simple
Modern programming language
Object oriented
Type safe
Scalable and Updateable
Rich Library
Fast speed
C#

Section 3

VISUAL STUDIO

Integrated Development Environment

- Syntax highlighting
- Code completion
- Refactoring
- Version control
- Debugging
- Code search
- Visual programming
- Language support

Visual Studio

Develop

- Navigate, write, and fix your code fast

Extend

- Choose from thousands of extensions to customize your IDE

Debug

- Debug, profile, and diagnose with ease

Collaborate

- Use version control, be agile, collaborate efficiently

Test

- Write high-quality code with comprehensive testing tools

C# Version History

Version	.NET Framework	Visual Studio
C# 1.0	.NET Framework 1.0/1.1	Visual Studio .NET 2002
C# 2.0	.NET Framework 2.0	Visual Studio 2005
C# 3.0	.NET Framework 3.0/3.5	Visual Studio 2008
C# 4.0	.NET Framework 4.0	Visual Studio 2010
C# 5.0	.NET Framework 4.5	Visual Studio 2012/2013
C# 6.0	.NET Framework 4.6	Visual Studio 2013/2015
C# 7.0	.NET Core 2.0	Visual Studio 2017
C# 8.0	.NET Core 3.0	Visual Studio 2019

Section 3

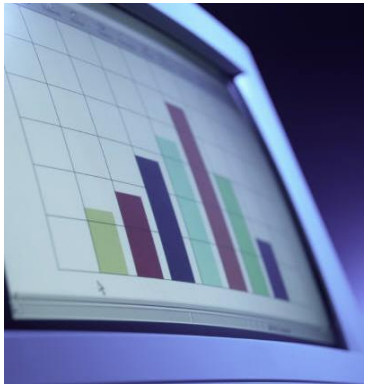
FIRST PROGRAM IN C# WITH VISUAL STUDIO

Hello World

- Create Hello World console application in Visual Studio
- Write first code
- Visual Studio windows
- Run application and check result

Input and Output

- Programmers often need to display the output of a C# program to users.
- The programmer can use the command line interface to display the output.
- The programmer can similarly accept inputs from a user through the command line interface.
- Such input and output operations are also known as console operations.



Console Operations

Console operations are tasks performed on the command line interface using executable commands.

The console operations are used in software applications because these operations are easily controlled by the operating system.

This is because console operations are dependent on the input and output devices of the computer system.

A console application is one that performs operations at the command prompt.

All console applications consist of three streams, which are a series of bytes. These streams are attached to the input and output devices of the computer system and they handle the input and output operations.

Console Operations

Standard `in`	• The standard in stream takes the input and passes it to the console application for processing.
Standard `out`	• The standard out stream displays the output on the monitor.
Standard `err`	• The standard err stream displays error messages on the monitor.

Basic I/O

- Console.Write command:
  - ✓ Writes the text representation of the specified value or values to the standard output stream
- Console.WriteLine command:
  - ✓ Writes the specified data, **followed by the current line terminator**, to the standard output stream

Basic I/O

- Console.Read command:
  - ✓ Reads the next character from the standard input stream.
- Console.ReadKey command:
  - ✓ Obtains the next character or function key pressed by the user
- Console.ReadLine command:
  - ✓ Reads the next line of characters from the standard input stream.
  - ✓ Finish input when user press the **Enter** key
  - ✓ Also used to pause screen

Lesson Summary

- Microsoft .NET Framework
- C# programming language
- Visual Studio
- Create first project: Hello World
- Basic I/O

Thank you