

INTRODUCTION TO .NET AND C#

10/22/2024

Chapter One

Contents

2

1. .NET Programming Framework & Languages
2. What is C#?
3. Visual design and event-driven programming
4. Places to write codes in C#

What is .NET?

3

- **Microsoft .NET** is a Framework which provides a **common platform** to Execute or, Run the applications developed in **various programming languages**.
 - Microsoft announced the .NET initiative in July 2000.
 - The main intention was to bridge the gap in **interoperability** between services of various programming languages
- .Net is not a programming language;
 - It supports **multiple programming languages** for different types of programmers.

- .NET framework also enables a programmer to create **sharable components** to be used in distributed application.
 - It is language independent platform.
- .Net framework supports OOPs model for multiple languages in a manner to allow language **interoperability**.

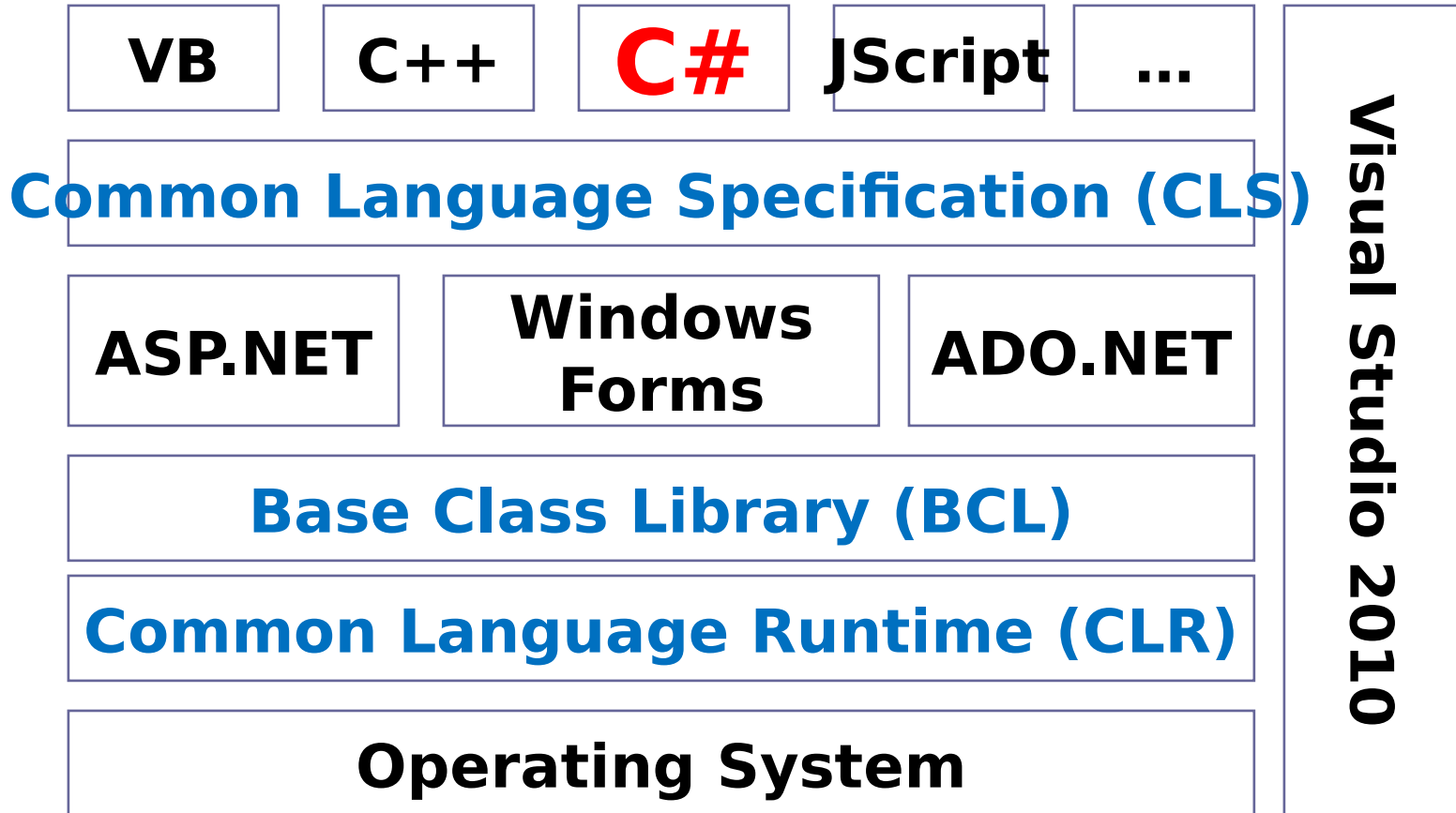
.NET Framework Objectives

5

- The .NET Framework is designed to fulfill the following objectives:
 - Provide **object-oriented programming** environment
 - Provide environment for **developing various types of applications**, such as
 - Windows-based applications and
 - Web-based applications
 - To ensure that code based on the .NET Framework can **integrate with any other code**

.NET Framework

6



The .NET Framework consists of:

7

- **The Common Language Specification (CLS)**
 - It contains guidelines, that language should follow so that they can communicate with other .NET languages. It is also responsible for Type matching.
- **The Framework Base Class Libraries (BCL)**
 - A consistent, object-oriented library of prepackaged functionality and Applications.
- **The Common Language Runtime (CLR)**
 - A **language-neutral** development & **execution environment** that provides **common runtime** for application execution .

Common Language Specification

8

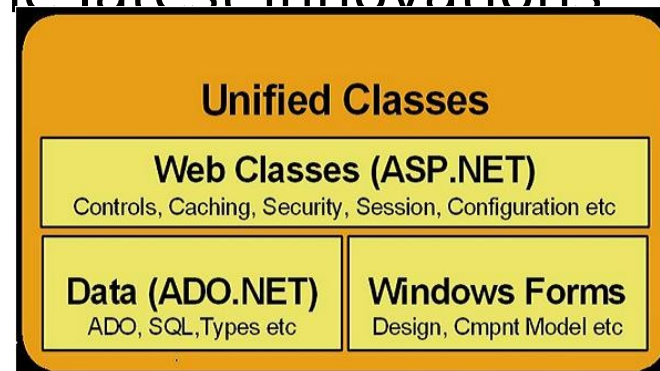
- CLS performs the following functions:
 - Establishes a framework that helps enable **cross-language integration, type safety,** and **high performance** code execution
 - Provides an **object-oriented model** that supports the complete implementation of many programming languages
 - **Defines rules that languages** must follow, which helps ensure that objects written in different languages can interact with each other

.NET Framework Base Class Library

9

- The Class Library is a **comprehensive, object-oriented collection** of reusable types
- The classes provide **a foundation of common functionality** and constructs that help **simplify application development** and Remove the requirement for developers to constantly reinvent logic.
- Allows to **implement applications quickly**

- The functionality of the base framework classes resides in the **namespace** called **System**
- These class library can be used to develop applications that include:
 - Traditional command-line applications
 - Graphical user interface (GUI) applications
 - Applications based on the latest innovations provided by ASP.NET
 - Web Forms
 - XML Web services



□ It provides the following:

- Input / Output operations
- String handling
- Managing arrays, lists, maps, etc
- Accessing files & file systems
- Accessing the registry
- Security
- Windowing
- Database management
- Drawing
- Managing errors & exceptions
- Connecting to Internet

Common Language Runtime (CLR)

12

- .NET Common Language Runtime (CLR), which is responsible for **maintaining the execution of all applications** developed using the .NET library.
- CLR provides **interoperability** between different language, like C# , VB, Visual C++, by providing a common environment for the execution of code written in of these languages.
- CLR **provides environment** for all .NET languages compilers for **converting source code into a common language** known as **IL** or **MSIL** or **CIL**.

- In .NET Framework's each compiler converts source code into an intermediate code is known as **Microsoft Inter-mediate language code**.
 - The advantage of **MSIL** code is: **it can be consumed by any other .NET compatible language**.
 - The MSIL code is **common for all languages** and is understandable only to .NET environment.
- When you compile code that uses the .NET Framework library, you don't immediately create operating-system-specific native code.

- Instead, you compile your code into **Common Intermediate Language (CIL) code**. This code isn't specific to any operating system (OS) and isn't specific to C#.
 - Obviously, more work is necessary to execute an application. That is the job of a **just-in-time (JIT)** compiler, which compiles CIL into native code that is specific to the OS and machine architecture being targeted.
- Only at this point can the OS execute the application. The just-in-time part of the name reflects the fact that **CIL code is compiled only when it is needed**.

- When you compile an application, the compiler generates an **executable file that the CLR can run**. This file is called an **assembly**.
 - An assembly contains code in an **intermediate format** called Microsoft intermediate language (**MSIL**).
 - All compilers for the .NET Framework generate code in this format, regardless of the programming language that was used to write an application.
- Assemblies are the building blocks of .NET Framework applications;
 - they form the fundamental unit of deployment, version control, reuse, and security.

□ CLR ensures:

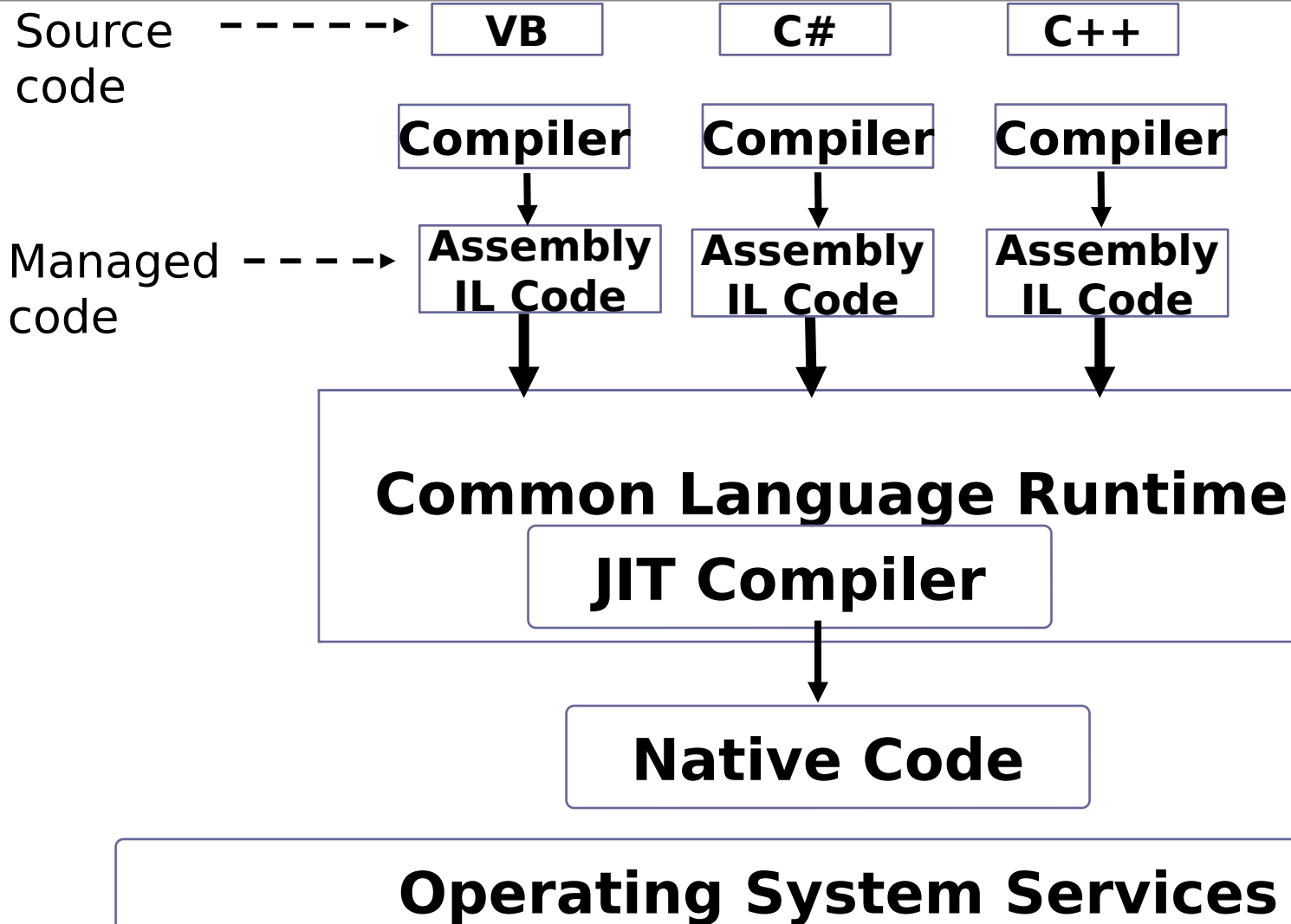
- A common runtime environment for all .NET languages
- Uses Common Type System (strict-type & code-verification)
- Memory allocation and garbage collection
- Intermediate Language (IL) to native code compiler. Which Compiles MSIL code into native executable code
- Security and interoperability of the code with other languages

□ **Over 36 languages supported today**

- C#, VB, Jscript, Visual C++ from Microsoft,
- Perl, Python, Smalltalk, Cobol, Haskell, Mercury, Eiffel, Oberon, Oz, Pascal, APL, CAML, Scheme, etc.

Execution in CLR

17



Contents

18

1. .NET Programming Framework & Languages
2. **What is C#?**
3. Visual design and event-driven programming
4. Places to write codes in C#

C# (C Sharp)

19

- Microsoft C# (pronounced C Sharp) developed by Microsoft Corporation, USA
 - New programming language that runs on the .NET Framework
 - C# is simple, modern, type safe, and object oriented
 - C# code is compiled as managed code (Code that targets the CLR)
 - Combines the best features of Visual Basic, C++ and Java

Applications you can write with C#

20

- Windows applications
- Web applications: ASP.NET, ASP.NET MVC
- Web services :- cloud service
- Service:- WCF(windows communication Foundation):- is a framework for building service- oriented application.

Characteristics of C#

21

□ **Simple**

- C# simplifies C++ by eliminating some operators such as `->`, `::`, and pointers.
- C# treats integers & Boolean data types as entirely different types.

□ **Consistent**

- C# supports an unified type system which eliminates the problem of varying ranges of integer types.
- All types are treated as objects.

10/22/2024

□ **Modern**

- C# is called modern language it supports:
- Automatic garbage collection
- Rich intrinsic model for error handling
- Decimal data types for financial applications
- Modern approach to debugging
- Robust security model

□ **Object-Oriented**

- C# is truly **object-oriented**. It supports:
 - Encapsulation
 - Inheritance
 - Polymorphism

□ **Compatible**

- C# enforces the .NET common language specifications & therefore allows interoperability with other .NET languages

□ **Interoperable**

- C# provides support for using COM objects, no matter what language was used to author them.

□ **Type-Safe**

- Type safety promotes robust programs.
- C# incorporates number of type-safe measures:
- All dynamically allocated objects & arrays are initialized to zero
- Use of any uninitialized variables produces an error message by the compiler
- Access to arrays are range-checked
- C# does not support unsafe casts
- C# supports automatic garbage collection

C# Language - Example

25

```
using System;

class HelloEthiopia
{
    public static void Main()
    {
        Console.WriteLine("Hello, Ethiopia!");
    }
}
```

Questions?

Thank You