

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Information Technology

C#.Net

By Prof. J. V. Mashalkar

Unit I Chapter 1

Introduction to .net

- What is .Net?
- .Net Framework
- CLR
- Visual Studio.Net & .Net Languages
- Integrated Development Environment
- Project types
- C#.Net history & design Goals
- How C# differs from C++
- Characteristics of C#.net

Introduction to .Net or What is .Net?

- Microsoft .Net is a software component that runs on Windows Operating system
- .Net provides tools and libraries that enable the developers to build Windows applications faster and easier.
- The .Net platform is an integral component of the Microsoft Windows for building and running next generation software applications and web services.

Advantages of .Net

- Multilanguage support
- Automatic resource management
- Simplification of application deployment
- Security
- Object oriented model

.Net Framework

- It is an application execution environment that is ideal for the development, deployment and execution of applications on Windows platform.

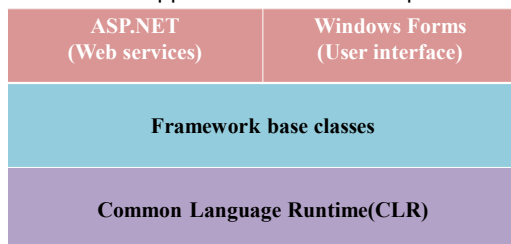


Figure: Architecture of .Net framework

These are:

- 1.Common Language Runtime (CLR)
- 2.Framework base classes or Base Class Library (BCL)
3. User and program interface

1.Common Language Runtime(CLR)

- It is responsible for loading and running all .net programs
- It is fully protected from the outside environment and it is highly optimized
- C#.net program → C#.net compiler→ MSIL code→CLR(Just-In-Time compiler)→Native or machine code

- Language compiler (Ex, compiler of C# or VB.Net) will convert the code to Microsoft Intermediate Language (MSIL) in turn this will be converted into native code by CLR.

Functions of CLR

- Loads and executes program
- Converts MSIL code to native machine code
- Isolates memory for applications
- Manages memory and objects (automatic garbage collector)
- Manages exceptions
- Provides type checking
- Provides code metadata

CLR includes:

- a. Common Type System(CTS)
- b. Common Language Specification(CLS)
- c. Microsoft Intermediate Language(MSIL)
- d. Managed Code

a. Common Type System(CTS)

- The .Net framework supports multiple languages by using CTS.
- CTS defines a set of data types that can be used in common.
- It ensures that objects written in different languages can interact with each other

The CTS supports two categories of types. These are:

1) Value types

- Ex. int, float, double etc.

2) Reference types

- Ex. class, interface, delegate, string etc.

b. Common Language Specification(CLS)

- The CLS defines a set of rules that enable interoperability on the .net platform.
- The CLS is a subset of CTS and therefore the languages supporting CLS can use each other class libraries as if they are their own.

c. Microsoft Intermediate Language(MSIL)

- MSIL is an instruction set into which all .Net programs are compiled.
- It contains instructions for **loading, storing, initializing and calling methods.**

JIT compilation:

- Before executing on the target machine, MSIL has to be translated into the machines native code.
- This can either be done before the application is called or at runtime.
- At runtime, the translation is done by a **Just-In-Time Compiler(JIT)**

d. Managed code

- Data that is under the control of the CLR garbage collection process is called managed code. OR
- The code that satisfies the CLR at run time is called as managed code.

2. Framework base classes or Base Class Library

- It is also known as Base Class Library(BCL)
 - .Net framework class library (FCL) is a set of managed classes that provide access to system services. H
- Ex. Input/output, database access etc.
- It provides a rich collection of classes.
 - These managed classes of FCL are divided into separate namespaces based on functionality.

- Example: classes used for local pertaining input/output can be found in the namespace System.
- The developers can use these classes to develop their applications quickly by:
 - invoking them or
 - inheriting them to derived classes for extending their functionality.

System.Console.WriteLine("Hello");

Namespace name Class name Method name

3.User and program interfaces

The .Net framework provides the following tools for managing user and application interfaces:

i) Windows forms:
Ex. Windows forms (C#.Net , VB.Net) is the form generator for client-side applications.

ii) Console applications:
Ex. Console applications are useful for logging or other text based activities.

iii) Web Forms:
Ex. ASP.Net is used to create dynamic Web applications.

- These tools enable users to develop user friendly **desktop based** as well as **web** based applications using a wide variety of languages on the .Net platform.

.Net Languages

We can use number of languages for developing .Net applications .These are:

- Visual Basic
- Visual C++
- Visual C#

History of C#

The first version of C# .Net was developed by **a team under the leadership of Anders Hejlsberg at Microsoft in 2000.**

• Project types:

Project type	description
Class Library	-A class library is a basic code building component which can not be executed and it does not have visible interface -It contains component & libraries to be used inside other applications.
Windows Control Library	-If the controls that come with Visual Studio don't provide the functionality you need, you can built your own custom control. -It contains user defined windows control to be used by windows application

Project type	Description
Console application	-A console application is an application with a very limited user interface. -This type of application displays its output on a command window and receive input from the same window.
Windows Forms Application	A windows forms application is used to develop an user interactive application for database handling, for creating calculator, for creating applications like Wordpad, Notepad etc.

Project type	Description
Windows Service	These applications are the long running application that don't have a visible interface. These services can be started automatically when the computer is turned on, paused and restarted.
ASP.Net web application	Web applications are the programs that can be run inside some web server(e.g. IIS-Internet Information Server) to fulfill user requests.
ASP.Net web service	Web services are the web applications that provide services to other applications over the internet.

Project type	Description
Web control library	Web control library contained user defined web controls to be used by web applications.

How C# differs from C++

C++	C#
C++ is a middle level language	C# is an object oriented language
When compiled, C++ code is converted into assembly code C++ program → C++ compiler → Assembly code → Assembler → Native code	When compiled, C#.net code is converted into intermediate language code. This intermediate code is converted into native code through the process called Just-In-Time Compilation
In C++, the memory i.e. allocated in the heap dynamically has to be explicitly deleted	In C#, memory management is automatically handled by garbage collector

C++	C#
In C++ switch statement, test variable can not be a string	In C# switch statement, the test variable can be a string
In C++ switch statement, when break statement is not given the fall-through will happen to the next case statement even if the current case statement has any code	In C#, switch statement when break statement is not given, the fall through will not happen to the next case statement if the current statement has any code.
C++ does not contain foreach statement	In addition to for, while and do_while, C# has another flow control statement called foreach.

C++	C#
In C++, the end of the class definition has a closing brace } followed by a semicolon.	In C#, the end of the class definition has a closing brace} only
In C++, the access specifiers are public, private and protected.	In C#, access specifiers are public, private, protected, internal and protected internal.
C++ does not have finally block in exception handling	C# has finally block in exception handling. The code statement in the finally block will be executed irrespective of exception occurrence.
C++ has the concept of function pointer	C# does not have the concept of function pointer. C# has a similar concept called delegate.

- **Characteristics of C#.Net**
1. Simple
 2. Consistent
 3. Modern
 4. Object oriented
 5. Type safe
 6. Versioning
 7. Compatible

1. Simple

- C# simplifies C++ by eliminating complex operators like `::`, `→`, `*`, `::*` etc. and pointer concept.
- C# treats the data types as per their use and range.

2. Consistent

- C# supports a **Common Type System** which eliminates the problem of varying ranges of data types.
- It also allows the developers to extend the type system.

3. Modern

- C# is called a modern language due to a number of features it supports. It supports:
 - Automatic garbage collection
 - Rich model for error handling
 - Modern approach to debugging
 - Robust security model

4. Object oriented

- C# is truly object oriented .It supports all the three concepts of object oriented systems:
 - Encapsulation
 - Inheritance
 - Polymorphism

5. Type safe

- C# reports a number of type safe measures on dynamically allocated objects
- Use of **any uninitialized variable** produces an error message by the compiler
- Access to **arrays are range checked** and warned if it goes out of bound
- It enforces **overflow checking** in arithmetic operations

6. Versioning

- Making new versions of software modules to work with the existing application is known as versioning.
- **C# provides support for versioning.**

7. Compatible

C# enforces the .Net **Common Language Specification(CLS)** and therefore allows interoperation with other .Net languages