MASTER OF COMPUTER APPLICATIONS (MCA)

PG Department of Computer Science and Technology

Sardar Patel University

Vallabh Vidyanagar, Gujarat, INDIA

PS02CMCA54 [The .NET Framework]

MR. BHARAT B. PATEL (ASSOCIATE PROFESSOR)

(✉) bb_patel@spuvvn.edu

The .NET Technology

- .NET Framework is a software development platform developed by Microsoft for building and running different types of applications.
- The .NET framework consists of developer tools, programming languages, and libraries to build desktop and web applications.
- It is also used to build websites, web services, and games, etc.
- .NET is a free, cross-platform, open source developer platform for building many different types of applications.
- With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, and IoT.

The .NET Technology

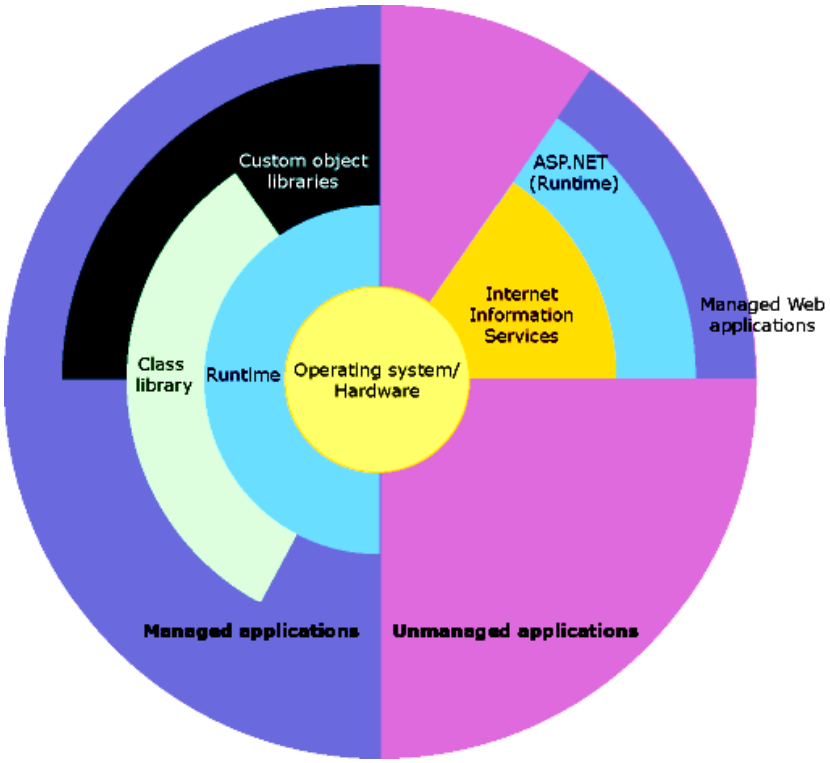
- **The .NET Framework (.NET is pronounced "dot net") is a set of technologies for developing and running software.**
- **The .NET Framework (pronounced dot net) is a software framework that runs primarily on Microsoft Windows.**
- **The .NET Framework is an application development platform that provides services for building, deploying, and running desktop, web, and phone applications and web services.**
- **There are two main components of the .NET Framework :**

  [1] CLR (COMMON LANGUAGE LIBRARY) : CLR is an execution environment, providing memory, process, thread management and other system services.

The .NET Technology

[2] BCL (BASE CLASS LIBRARY) : BCL (sometime known as Framework Class Library - FCL) is a comprehensive, object-oriented collection of reusable types that developers can use to create software. BCL includes tested, reusable code for all major areas of application development.

.NET Framework in Context



.NET Framework - Introduction

- The .NET Framework consists of the common language runtime and the .NET Framework class library.
- The common language runtime is the foundation of the .NET Framework.
- You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness.
- In fact, the concept of code management is a fundamental principle of the runtime.
- Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

.NET Framework - Introduction

- **The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.**
- **The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features.**

.NET Framework - Introduction

- The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.
- For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code.
- ASP.NET works directly with the runtime to enable ASP.NET applications and XML Web services, both of which are discussed later in this topic.
- Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension).

.NET Framework - Introduction

- **Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents.**
- **Hosting the runtime in this way makes managed mobile code possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and isolated file storage.**

.NET Framework - Introduction

Objectives :

- **To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.**
- **To provide a code-execution environment that minimizes software deployment and versioning conflicts.**
- **To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.**
- **To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.**

.NET Framework - Introduction

Objectives :

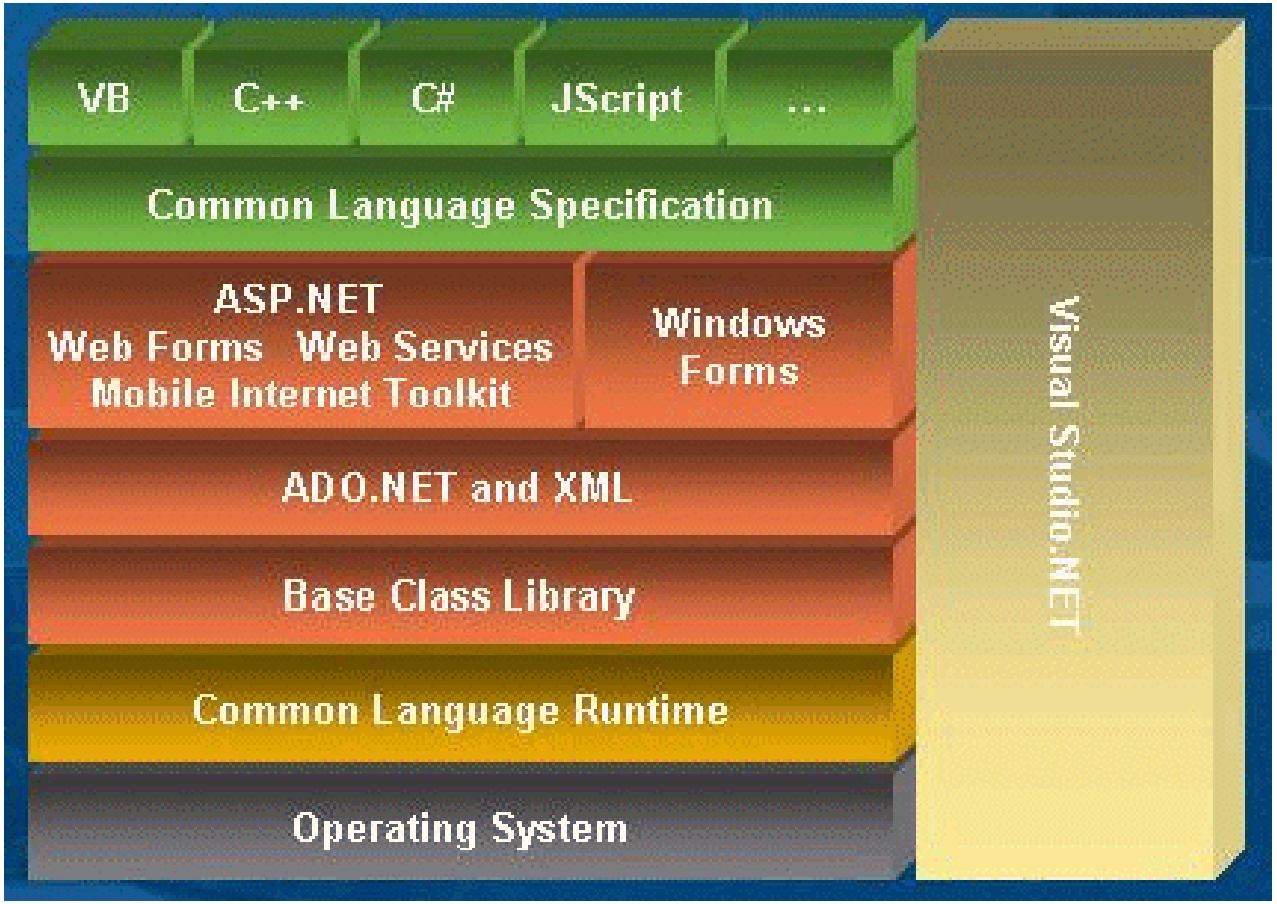
- **To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.**
- **To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.**

.NET Framework - Services

Services :

- **NET Framework provides the following services:**
- **Tools for developing software applications ,**
- **run-time environments for software application to execute,**
- **server infrastructure,**
- **value added intelligent software which helps developers to do less coding and work efficiently.**

.NET Framework - Architecture



.NET Framework - History

Overview of .NET Framework release history[2][3]

Version number	CLR version	Release date	Support ended	Development tool	Included in	
					Windows	Windows Server
1.0	1.0	2002-02-13	2009-07-14[4]	Visual Studio .NET[5]	XP[a]	N/A
1.1	1.1	2003-04-24		Visual Studio .NET 2003[5]	N/A	2003
2.0	2.0	2005-11-07	2011-07-12[4]	Visual Studio 2005[7]	N/A	2003, 2003 R2,[8] 2008 SP2, 2008 R2 SP1
3.0	2.0	2006-11-06	2011-07-12[4]	Expression Blend[9][b]	Vista	2008 SP2, 2008 R2 SP1
3.5	2.0	2007-11-19	2011-07-12[4] (except 3.5 SP1)	Visual Studio 2008[10]	7, 8[c], 8.1[c], 10[c]	2008 R2 SP1
4.0	4	2010-04-12	2016-01-12[11]	Visual Studio 2010[12]	N/A	N/A
4.5	4	2012-08-15	2016-01-12[11]	Visual Studio 2012[13]	8	2012
4.5.1	4	2013-10-17	2016-01-12[11]	Visual Studio 2013[14]	8.1	2012 R2
4.5.2	4	2014-05-05	N/A	N/A	N/A	N/A
4.6	4	2015-07-20	N/A	Visual Studio 2015[15]	10	N/A
4.6.1	4	2015-11-30[16]	N/A	Visual Studio 2015 Update 1	10 v1511	N/A
4.6.2	4	2016-08-02[17]	N/A		10 v1607	2016
4.7	4	2017-04-05[18]	N/A	Visual Studio 2017	10 v1703	N/A

CLR (Common Language Runtime)

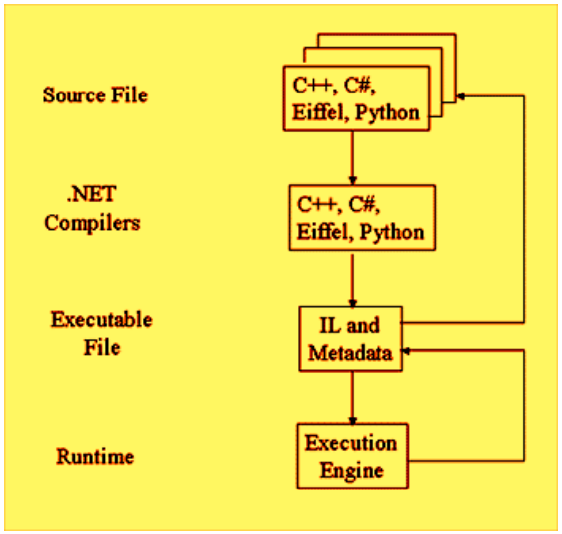
- The **Common Language Runtime (CLR)** is a special run time environment that provides the underlying infrastructure for Microsoft's .NET framework.
- This runtime is where the source code of an application is compiled into an intermediate language called CIL , originally known as **MSIL (Microsoft Intermediate Language).**
- When the program is then run, the CIL code is translated into the native code of the operating system using a **Just-In-Time (JIT)** compiler.
- The CLR is the engine behind the .NET Framework that handles the execution of all managed application code.
- CLR plays an intermediation role between the operating system and .NET applications.

CLR (Common Language Runtime)

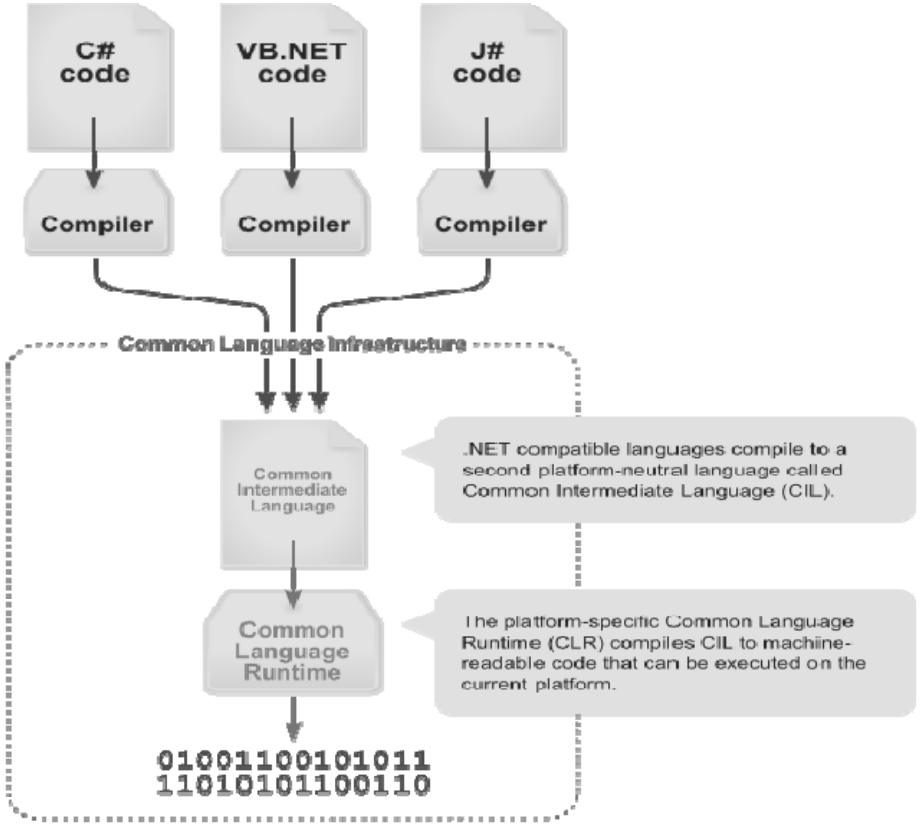
- Applications developed in the .NET platform don't directly interact with the operating system. Rather, they talk to the CLR and the CLR manages/handles the execution of the applications.
- When we write a program using any of the .NET languages like C#, VB or C++, the source code doesn't get compiled to machine code.
- Each language specific compiler (VB compiler, C# compiler etc) converts the source code to an intermediate language code which is known as IL (Intermediate Language) code.

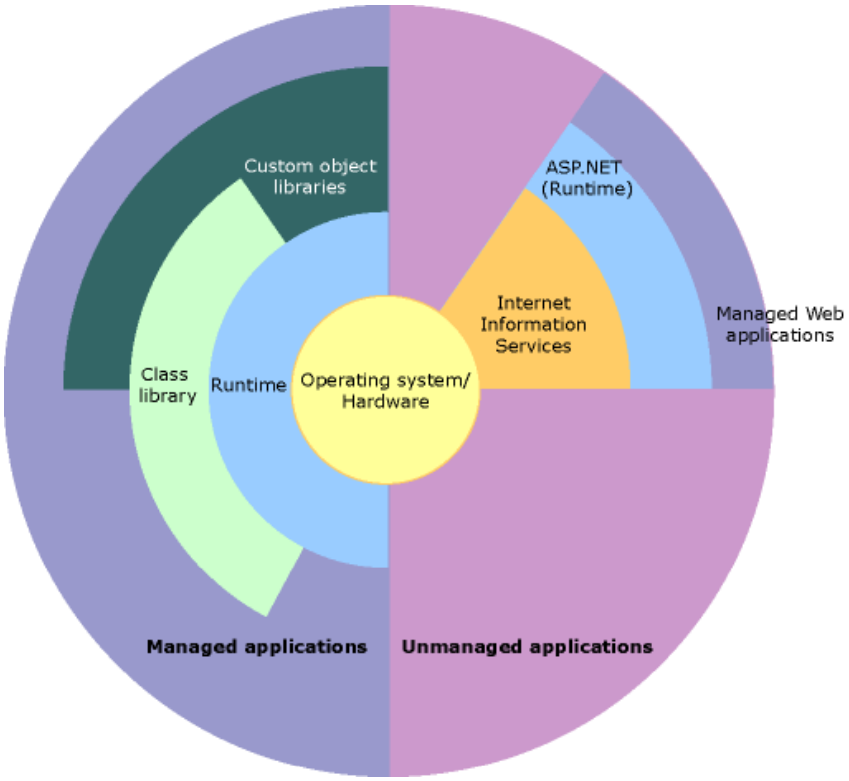
CLR (Common Language Runtime)

- The IL code then is converted to machine specific code by the CLR engine at runtime.
- The CLR is responsible for managing memory allocation, starting and terminating threads and processes and enforcing security policy at runtime.



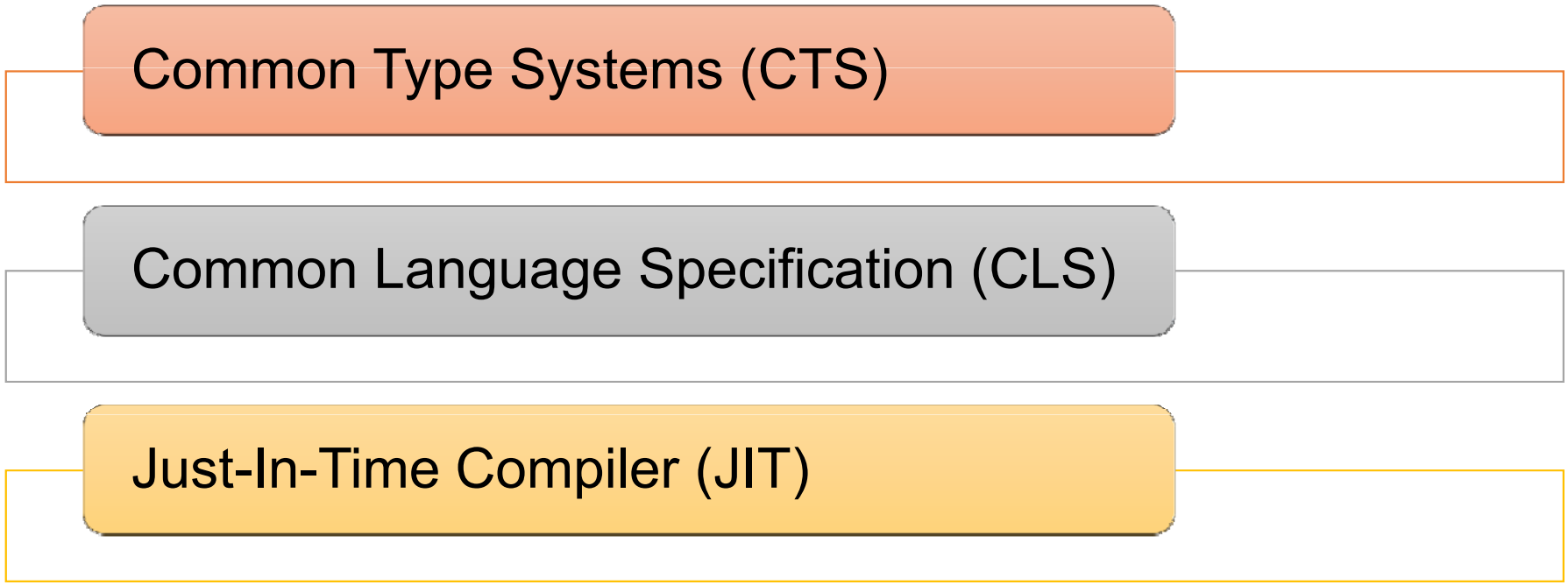
Common Language Infrastructure





Components of CLR (Common Language Runtime)

- **Components of CLR are :**



Benefits of CLR (Common Language Runtime)

- **Performance improvements.**
- **The ability to easily use components developed in other languages.**
- **Extensible types provided by a class library.**
- **New language features such as inheritance, interfaces, and overloading for object-oriented programming; support for explicit free threading that allows creation of multithreaded, scalable applications; support for structured exception handling and custom attributes.**

BCL (Base Class Library)

- **The BCL provides the system functionality in the .NET Framework as it has various classes, data types, interfaces, etc. to perform multiple functions and build different types of applications such as desktop applications, web applications, mobile applications, etc.**
- **The BCL aims to provide general implementations without bias to any workload.**
- **Performance is an important consideration, since apps might prefer a particular policy, such as low-latency to high-throughput or low-memory to low-CPU usage.**
- **The BCL is intended to be high-performance generally, and take a middle-ground approach according to these various performance concerns.**
- **The BCL is integrated with the Common Language Runtime (CLR) of the .NET framework and is used by all the .NET languages such as C#, F#, Visual Basic .NET, etc.**

Namespaces in BCL

- **Namespaces in the Base Class Library are a group of related classes and interfaces that can be used by all the .NET framework languages.**
- **Namespaces in C# are used to organize too many classes so that it can be easy to handle the application.**

Namespaces in BCL

- Some of the namespaces in the FCL along with their description is given as follows:

Namespace	Description
System	The System namespace has base classes for definition of interfaces, data types, events, event handlers, attributes, processing exceptions etc.
System.Collections	The System.Collections namespace has multiple standard, specialized, and generic collection objects that are defined using various types.
System.Data	The System.Data namespace accesses and manages data from various sources using different classes.
System.Drawing	The System.Drawing namespace handles GDI+ basic graphics functionality. Various child namespaces also handle vector graphics functionality, advanced imaging functionality, etc.
System.IO	The System.IO namespaces support IO like data read/write into streams, data compression, communicate using named pipes etc. using various types.
System.Linq	The System.Linq namespace supports Language-Integrated Query (LINQ) using various types.
System.Net	The System.Net namespace provides an interface for network protocols, cache policies for web resources, composing and sending e-mail etc. using various classes.
System.Security	The System.Security namespace has the .NET security system and permissions. Child namespaces provide authentication, cryptographic services etc.
System.Threading	The System.Threading namespace allows multithreaded programming using various types.

.NET Languages

- **C# (pronounced "C sharp") is a simple, modern, object-oriented, and type-safe programming language.**
- **Its roots in the C family of languages makes C# immediately familiar to C, C++, Java, and JavaScript programmers.**

.NET Languages

- **C# - Microsoft's flagship .NET Framework language which bears similarities to the C++ and Java languages.**
- **Visual Basic .NET - A completely redesigned version of the Visual Basic language for the .NET Framework. This also includes Visual Basic 2005 (v8.0).**
- **VBx, a dynamic version of Visual Basic .NET that runs on top of the Dynamic Language Runtime.**
- **C++/CLI and the deprecated Managed C++ - A managed version of the C++ language.**
- **J# - A Java and J++ .NET transitional language.**
- **JScript .NET - A compiled version of the JScript language.**
- **Windows PowerShell - An interactive command line shell/scripting language which provides full access to the .NET Framework.**

.NET Languages

- **IronPython - A .NET implementation of the Python programming language developed by Jim Hugunin at Microsoft.**
- **IronRuby - A dynamically compiled version of the Ruby programming language targeting the .NET Framework.**
- **F#, a member of the ML programming language family.**