**Introduction To C# (C-Sharp) Programming**

Microsoft introduced a new language called C# (pronounced C Sharp). C# is designed to be a simple, modern, general-purpose, object-oriented programming language, borrowing key concepts from several other languages, most notably Java.

C# was developed by **Anders Hejlsberg** and his team during the development of **.Net Framework.**

C# is a part of .NET Framework.

Microsoft introduced C# as a new programming language to address the problems posed by traditional languages.

**We can use C# for building variety of applications**

* **WINDOWS APPLICATION**: using console application or winform application.
* **MOBILE APPLICATIONS:** for phones such as Nokia Lumia (built-in support) but we can use a third party tool or library called “**XAMARIN**” to create mobile applications for ANDROID and IOS as well.
* **WEB APPLICATON:** using ASP.NET web forms or ASP.NET MVC.
* **GAMING APPLICATION:** Unity.

C# could theoretically be compiled to machine code, but in real life, it's always used in combination with the .NET framework. Therefore, applications written in C#, requires the .NET framework to be installed on the computer running the application. While the .NET framework makes it possible to use a wide range of languages, C# is sometimes referred to as THE .NET language, perhaps because it was designed together with the framework.

**Language interoperability** is the ability of code to interact with code that is written by using a different programming language.

**What is .Net Framework ?**

* .NET is a programming framework created by Microsoft that developers can use to create applications more easily. A framework is just a bunch of code that the programmer can call without having to write it explicitly.
* It provides a controlled programming environment where software can be developed, installed and executed on Windows-based operating systems.
* It is basically a collection of libraries.
* Is a programming platform that is used for developing Windows, Web-based, and mobile software.
* It has a number of pre-coded solutions that manage the execution of programs written specifically for the framework.
* A programmer can develop applications using one of the languages supported by .NET.
* Microsoft introduced C# as a new programming language to address the problems posed by traditional languages.
* The .NET Framework is an infrastructure that is used to build, deploy, and run different types of applications and services using .NET technologies.
* The .NET Framework is an infrastructure that is used to Minimize software development, deployment, and versioning conflicts.

**Microsoft C# Was Developed To**

* Create a very simple and yet powerful tool for building interoperable 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# has features common to most object-oriented languages.
* Provide consistent object-oriented programming environment.
* Minimize software deployment and versioning conflicts by providing a code-execution environment.
* Promote safe execution of code by providing a code-execution environment.
* Provide a consistent developer experience across varying types of applications such as Windows-based applications and Web-based applications.
* It has language-specific features, such as:

o Type safety checking  
o Generics  
o Indexers

* These features make the C# as a preferred language to create a wide variety of applications.
* C# is a programming language designed for building a wide range of applications that run on the .NET Framework.

**Purpose of C# Language**

* Microsoft .NET was formerly known as Next Generation Windows Services (NGWS).
* It is a completely new platform for developing the next generation of Windows/Web applications.
* These applications transcend device boundaries and fully harness the power of the Internet.
* However, building the new platform required a language that could take full advantage.
* This is one of the factors that led to the development of C#.
* C# is an object-oriented language derived from C and C++.
* The motive of C# is to provide a simple, efficient, productive, and object-oriented language that is familiar and yet at the same time revolutionary.

**Features of C#**

* C# is a programming language designed for building a wide range of applications that run on the .NET Framework.
* Following are some basic key features of C#:
  + Type-safety Checking
  + Object-oriented Programming
  + Garbage Collection
  + Standardization by European Computer Manufacturers Association (ECMA)
  + Generic Types and Methods
  + Iterators
  + Methods with named Arguments
  + Methods with optional Arguments
  + Static Classes
  + Nullable Types
  + Auto-implemented Properties
  + Accessor Accessibility
  + Anonymous Methods
  + Parallel Computing
  + Auto-implemented Properties
  + Partial Classes
* Object-oriented Programming: Focuses on objects so that code written once can be reused. This helps reduce time and effort on the part of developers.
* Type-safety Checking: Checked the overflow of types because uninitialized variables cannot be used in C# as C# is a case-sensitive language
* Garbage Collection: Performs automatic memory management from time to time and spares the programmer the task.
* Standardization by European Computer Manufacturers Association (ECMA): Specifies the syntax and constraints used to create standard C# programs.
* Generic Types and Methods: Are a type of data structure that contains code that remains the same throughout but the data type of the parameters can change with each use.
* Iterators: Enable looping (or iterations) on user-defined data types with the for each loop.
* Static Classes: Contain only static members and do not require instantiation.
* Partial Classes: Allow the user to split a single class into multiple source code (.cs) files.
* Anonymous Methods: Enable the user to specify a small block of code within the delegate declaration.
* Methods with named Arguments: Enable the user to associate a method argument with a name rather than its position in the argument list.
* Methods with optional Arguments: Allow the user to define a method with an optional argument with a default value.
* Nullable Types: Allow a variable to contain a value that is undefined.
* Accessor Accessibility: Allows the user to specify the accessibility levels of the get and set accessors.
* Auto-implemented Properties: Allow the user to create a property without explicitly providing the methods to get and set the value of the property.
* Parallel Computing: Support for parallel programming using which develop efficient, fine-grained, and scalable parallel code without working directly with threads or the thread pool.

**There are Several Applications Of C#**

**C# is an object-oriented language that can be used in a number of applications.**

* Gaming applications
* Web applications
* Mobile applications for pocket PCs, PDAs, and cell phones
* Web services
* Cloud applications
* Simple standalone desktop applications such as Library Management Systems, Student Mark Sheet generation, and so on
* Large-scale enterprise applications
* Complex distributed applications that can spread over a number of cities or countries

**The CLR**

* Is the foundation of the .NET Framework.
* Acts as an execution engine for the .NET Framework.
* Manages the execution of programs and provides a suitable environment for programs to run.
* Provides a multi-language execution environment.
* Is a backbone of .NET Framework
* Performs various functions such as:
  + Memory management
  + Code execution
  + Error handling
  + Code safety verification
  + Garbage collection

**By Using DotNet Framework**

* A programmer can develop applications using one of the languages supported by .NET.
* These applications make use of the base class libraries provided by the .NET Framework.
* The .NET Framework supports a number of development tools and language compilers in its Software Development Kit (SDK).

**For Example**

* To display a text message on the screen, the following command can be used:
* System.Console.WriteLine(".NET Architecture");
* The same WriteLine() method will be used across all .NET languages.
* This is done by making the Framework Class Library as a common class library for all .NET languages.

**DotNet Framework Class Library (FCL)**

* Is a comprehensive object-oriented collection of reusable types.
* Is used to develop applications ranging from traditional command-line to Graphical User Interface (GUI) applications that can be used on the Web.

**DotNet Framework Components**

* Common Language Specification (CLS)
* Common Type System (CTS)
* Base Framework Classes
* WPF
* Asp.Net
* WCF
* Ado.Net
* Entity Framework
* Task Parallel Library
* LINQ
* Parallel LINQ

The .NET Framework languages, such as C#, VB, and J# are statically typed languages.

**CLS - Common Language Specification**

* Is a set of rules that any .NET language should follow to create applications that are interoperable with other languages.

**CTS - Common Type System**

* Describes how data types are declared, used, and managed in the runtime and facilitates the use of types across various languages.

**ADO.NET**

* Provides classes to interact with databases.

**ASP.Net**

* Provides a set of classes to build Web applications. ASP.NET Web applications can be built using Web Forms, which is a set of classes to design forms for the Web pages similar to the HTML.
* Supports Web services that can be accessed using a standard set of protocols.

**WCF**

* Is a service-oriented messaging framework.
* Allows creating service endpoints and allows programs to asynchronously send and receive data from the service endpoint.

**WPF**

* Is a UI framework based on XML and vector graphics.
* Uses 3D computer graphics hardware and Direct3D technologies to create desktop applications with rich UI on the Windows platform.

**LINQ**

* Is a component that provides data querying capabilities to a .NET application.

**Entity Framework**

* Is a set of technologies built upon ADO.NET that enables creating data-centric applications in object-oriented manner.

**Parallel LINQ**

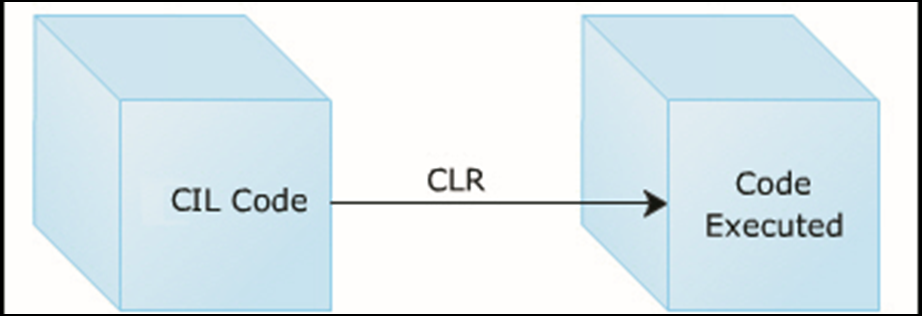
* Is a set of classes to support parallel programming using LINQ.

**Task Parallel Library**

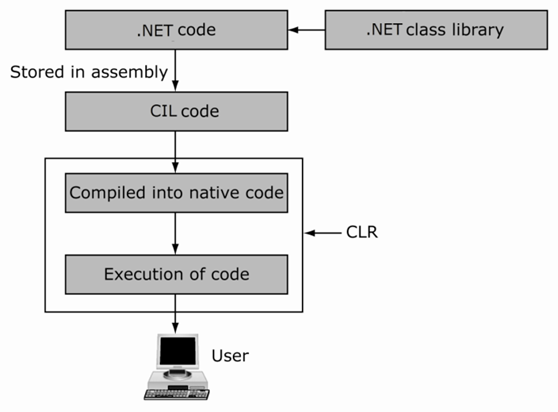
* Is a library that simplifies parallel and concurrent programming in a .NET application.

**When a code is executed for the first time**

* The CIL (COMMON INTERMEDIATE LANGUAGE) code is converted to a code native to the operating system.
* This is done at runtime by the Just-In-Time (JIT) compiler present in the CLR.
* The CLR converts the CIL code to the machine language code.
* Once this is done, the code can be directly executed by the CPU.



**The following figure represents the process of conversion of CIL code to the native code:**



**The CLR provides many features such as:**

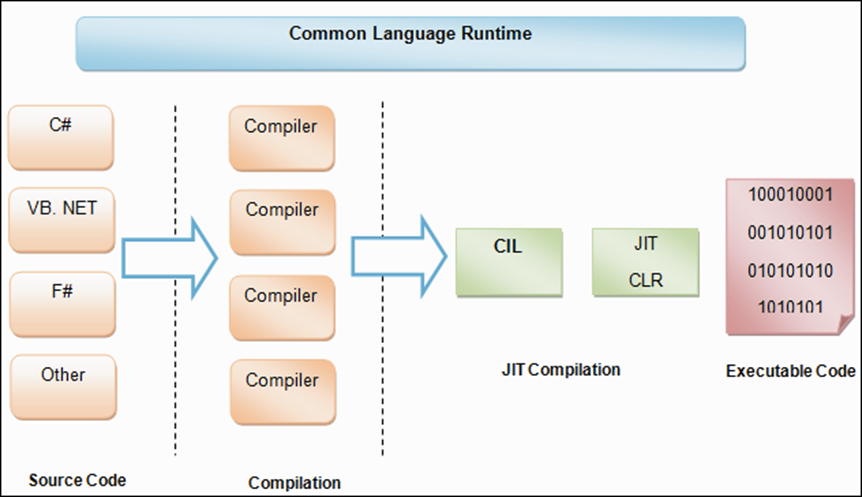
* Memory management
* Code execution
* Error handling
* Code safety verification
* Garbage collection

The applications that run under the CLR are called managed code.

The CLR manages code at execution time and performs operations such as:

* Thread management
* Remoting
* Memory management

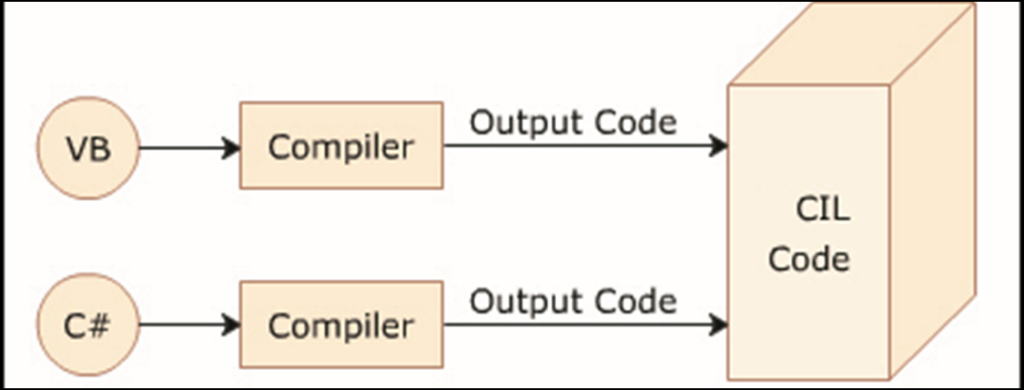
**Below figure shows a more detailed look at the working of the CLR:**



**CIL - Common Intermediate Language**

* Every .NET programming language generally has a compiler and a runtime environment of its own.
* The compiler converts the source code into executable code that can be run by the users.
* One of the primary goals of .NET Framework is to combine the runtime environments so that the developers can work with a single set of runtime services.
* When the code written in a .NET compatible language such as C# or VB is compiled, the output code is in the form of MSIL code.
* MSIL is composed of a specific set of instructions that indicate how the code should be executed.
* MSIL is now called as Common Intermediate Language (CIL).

**The following figure depicts the concept of CIL:**



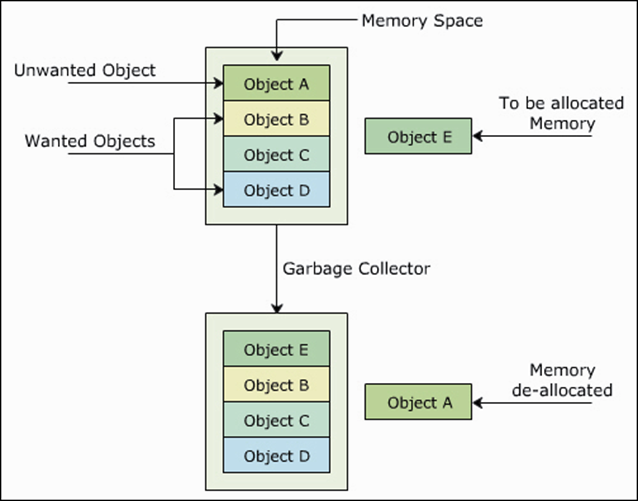
**What Is Memory Management ?**

* In programming languages like C and C++, the allocation and de-allocation of memory is done manually.
* Performing these tasks manually is both, time-consuming and difficult.
* The C# language provides the feature of allocating and releasing memory using automatic memory management.
* This means that there is no need to write code to allocate memory when objects are created or to release memory when objects are not required in the application.
* Automatic memory management increases the code quality and enhances the performance and the productivity.

**What Is Garbage Collection ?**

* Garbage Collection is the process of automatic reclaiming of memory from objects that are no longer in scope.
* Garbage Collection helps the process of allocating and de-allocating memory using automatic memory management.

**Below figure describes concept of garbage collection:**



**DLR - Dynamic Language Runtime**

* Is a runtime environment built on top of the CLR to enable interoperability of dynamic languages such as Ruby and Python with the .NET Framework.
* Allows creating and porting dynamic languages to the .NET Framework.
* Provides dynamic features to the existing statically typed languages. For example, C# relies on the DLR to perform dynamic binding.