Unmanaged Code

The code, which is developed outside .NET, Framework is known as unmanaged code.

Under the .NET Framework, however, this is a two - stage process. With help of MSIL and JIT.

MSIL (Microsoft Intermediate Language)

It is language independent code. 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 Microsoft Intermediate Language (MSIL) code. The MSIL code is not specific to any operating system or to any language.

JIT (Just-in-Time)

Just - in - Time (JIT) compiler, which compiles MSIL 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 MSIL code is only compiled as, and when, it is needed.

In the past, it was often necessary to compile your code into several applications, each of which targeted a specific operating system and CPU architecture.

This is now unnecessary, because JIT compilers (as their name suggests) use MSIL code, which is independent of the machine, operating system, and CPU. Several JIT compilers exist, each targeting a different architecture, and the appropriate one will be used to create the native code required.

Managed vs. Unmanaged Code

C# can only be used to build software that runs under the .NET framework. You can never use C# to build C or C++ style applications.
The code that runs under the Common Language Runtime (CLR) is managed code, or the code targeting the .NET runtime is managed code.
And the code is not hosted under the .NET framework or not targeted to the .NET framework is unmanaged code.
Managed code is stored in the binary unit called assembly. This environment makes it possible to build .NET programs on wide variety of machines.