**AKINTONWA ZAINAB AYOMIPO**

**EVOLUTION OF .NET FRAMEWORK AND C#**

C# and .NET were born of the need for a modern and powerful programming language and framework. In the late 1990s, Microsoft recognized the limitations of existing programming languages and set out to create a language that could leverage the full potential of the windows platform. Thus, C# and .NET came into existence.

C# was first introduced in 2000 as part of the .NET initiative. The C# programming language was designed by ANDERS HEJLSBERG from Microsoft in 2000 and was later approved as an international standard by Ecma (ECMA-334) in 2002. It was designed to be a simple, modern, and object oriented language that could address the challenges faced by developers. C# drew inspiration from various programming languages such as C++, JAVA, and DELPHI, combining their best features into a cohesive and elegant syntax. With the introduction of C#, Microsoft also released the .NET framework, a comprehensive platform for building windows-based applications. C# and .NET were a closed-source when it was first introduced.

.NET begins in 2002 with the release of version (**1.0)**.Back then, .NET was a closed framework accessible only through licensing. This approach had its drawbacks, it was not open source and developers could not peek under the hood of languages like C# or the .NET framework itself. However in 2012 Microsoft announced its intention to make .NET open source.

This decision not only thrilled the developer community but also instilled confidence in the future growth of .NET. Microsoft recognized the need to embrace this change and broaden its audience beyond corporations to cater developers seeking cost-effective solutions. It signified a shift in Microsoft mindset- a realization that the community’s input and contributions were invaluable, no longer was .NET an isolated entity.

In 2016 Microsoft introduced .NET Core, a cross-platform, open-source framework that enable developers to build application that could run on Windows, macOS, and Linux. .NET Core provided a lightweight and modular runtime, optimized for cloud based and containerized environments.

In 2020 a significant change took place as .NET 5 was introduced, dropping the “Core” and “Framework” distinctions from its name. While this caused some confusion initially, it was part of Microsoft broader plan.

The journey of C# and .NET has been a remarkable one, marked by continuous evolution and innovation. From its humble beginnings, C# has grown into a versatile and powerful language, enabling developers to create a wide range of applications. The .NET framework has provided a robust and scalable platform for application development, fostering a vibrant ecosystem.

.NET provides a large library of pre-coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework.

In summary, C# is a language and .NET is a framework, C# is commonly used to write code for the .NET framework.

**MONO:** this is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Microsoft, MONO is an open source implementation of Microsoft’s .NET framework as part of the .NET Foundation and based on the ECMA standards for C# and the Common Language Runtime. MONO includes a just-in-time (JIT) runtime engine for Windows, macOS, Linux and Android and compilers for C# and Visual Basic.

**XAMARIN:** is a developer’s tool for cross-platform mobile application development. It is for building mobile applications for Android, IOS, and Windows with C# and .NET framework.

**COM:** short for Component Object Model, it is a component software architecture from Microsoft. It defines a structure for building program routines (objects) that can be called up and executed in a Windows environment.

**.NET CORE:** a cross-platform, high performance, open-source framework for building modern, cloud-enabled, internet connected apps. It allows you to customize your application or website based on different languages and locales.

**UNITY C#:** a real time 3D development platform for building 2D and 3D applications. It offers a user-friendly interface and a powerful scripting language called C#, allowing developers to create games easily.

**REST: (**Representational State Transfer). A software architectural style that defines the set of rules to be used for creating web services.

**THREE KEY FUNCTIONS OF CLR ( Common Language Runtime)**

1. EXCEPTION HANDLING: the process of responding to the occurrence of exceptions during the execution of a program. Exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program.
2. MEMORY MANAGEMENT: the process of controlling and coordinating a computer’s main memory. It ensures that blocks of memory space are properly managed and allocated.
3. GARBAGE COLLECTION: a component of CLR that works as an automatic memory manager. The garbage collector manages the allocation and release of memory for an application so that developers working with managed code do not have to write code to perform memory management. It automatically free up memory space that has been allocated to objects no longer needed by the program.