**Name**: Sodiq Al Ameen Sunkanmi

**Department**: Computer Science

**Level**: HND1

**Course**: Com316

C# Assignment One

1. Write a short note on the evolution of .Net Framework and C# (100 words)
2. Explain the following terms; Mono, Xamarin, COM, .Net Core, Unity C#, REST
3. Critically explain any three key functions of CLR (50 words)

1) Microsoft started development on the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework) in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2001 the first beta versions of .NET Framework 1.0 were released. The first version of .NET Framework was released on 13 February 2002, bringing [managed code](https://en.wikipedia.org/wiki/Managed_code) to [Windows NT 4.0](https://en.wikipedia.org/wiki/Windows_NT_4.0), [98](https://en.wikipedia.org/wiki/Windows_98), [2000](https://en.wikipedia.org/wiki/Windows_2000), [ME](https://en.wikipedia.org/wiki/Windows_ME) and [XP](https://en.wikipedia.org/wiki/Windows_XP).

Since the first version, Microsoft has released nine more upgrades for .NET Framework, seven of which have been released along with a new version of [Visual Studio](https://en.wikipedia.org/wiki/Visual_Studio). Two of these upgrades, .NET Framework 2.0 and 4.0, have upgraded [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR). New versions of .NET Framework replace older versions when the CLR version is the same.

The .NET Framework family also includes two versions for [mobile](https://en.wikipedia.org/wiki/Mobile_computing) or [embedded device](https://en.wikipedia.org/wiki/Embedded_device) use. A reduced version of the framework, the [.NET Compact Framework](https://en.wikipedia.org/wiki/.NET_Compact_Framework), is available on [Windows CE](https://en.wikipedia.org/wiki/Windows_CE) platforms, including [Windows Mobile](https://en.wikipedia.org/wiki/Windows_Mobile) devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone" \o "Smartphone). Additionally, the [.NET Micro Framework](https://en.wikipedia.org/wiki/.NET_Micro_Framework) is targeted at severely resource-constrained devices.

.NET Framework 4.8 was announced as the final version of .NET Framework, with future work going into the rewritten and [cross-platform](https://en.wikipedia.org/wiki/Cross-platform_software) [.NET Core](https://en.wikipedia.org/wiki/.NET) platform (later, simply *.NET*), which shipped as *.NET 5* in November 2020. However, .NET Framework 4.8.1 was released in August 2022.

2) i) **Mono:** Mono is a free and open-source .NET Framework-compatible software framework.

ii) **Xamarin:** is a developer's tool for cross-platform mobile application development.

iii) **COM:** The Component Object Model (COM) lets an object expose its functionality to other components and to host applications on Windows platforms.

iv) **.Net Core**: is a new version of .Net Framework, which is a free, open source, general- purpose development platform maintained by Microsoft.

v) **Unity C#:** Unity is a real-time 3D development platform for building 2D and 3D application, like games and simulations, using .NET and the C# programming language.

vi) **REST:** Representational State Transfer (REST) allows application to interact with each other and exchange data.

### 3) Code safety verification

Code safety verification is an important security feature offered by the CLR. It verifies the code's integrity and type safety, preventing potentially harmful or malicious code from executing. The CLR performs various checks and validations on the code to ensure that it adheres to safety standards, mitigating security risks.

### Thread Executions

One of the key responsibilities of the CLR is managing thread executions. It provides a threading mechanism that allows multiple threads to run concurrently within an application. The CLR schedules and controls the execution of these threads, ensuring proper synchronization and coordination.

### Common Type System (CTS)

The Common Type System (CTS) is a fundamental aspect of the CLR. It defines a standardized set of data types that all .NET languages conform to. This ensures that data types used in different .NET languages are interoperable, allowing seamless integration and communication between components written in different languages.