**NAME: NGOZI DESTINY CHUKWUWINKE**

**Course code: Com 316**

**C# assignment 1**

A. Write a short note on the evolution of .Net Framework and C#(100 words)

**The \*\*.NET Framework\*\*** debuted in 2002, evolving to support Windows applications. Over time, it embraced web and cross-platform development. The introduction of \*\*C#\*\* in 2000 marked a modern, object-oriented language. Both underwent iterative updates, enhancing performance, security, and cross-platform compatibility. In 2016, .NET Core introduced a modular, open-source approach. The journey culminated in the unified \*\*.NET 5\*\* in 2020, merging .NET Core and Framework. Ongoing updates, like \*\*.NET 6\*\*, prioritize simplicity, performance, and a unified development platform.

**Overview of c# update**;

i. C# 1.0 (2000)

ii. C# 2.0 (2005)

Iii. C# 3.0 (2007)

iv. C# 4.0 (2010)

V. C# 5.0 (2012)

Vi. C# 6.0 (2015)

Vii. C# 7.0 (2017)

Viii. C# 8.0 (2019)

ix. C# 9.0 (2020)

B. Explain the following terms ;

Mono, Xamarin , COM , .Net Core,Unity C#,REST

**\*\*Mono\*\*** is an open-source, cross-platform implementation of the .NET Framework. Developed by Xamarin, Mono allows developers to build and run .NET applications on various operating systems, including Linux, macOS, and Windows. It provides a runtime environment, compilers, and libraries compatible with the .NET standards, enabling the development of cross-platform applications using languages such as C#. Mono has been utilized for a range of applications, including web development, mobile app development, and game development, contributing to the expansion of .NET beyond the Windows ecosystem.

**\*\*Com\*\*** commonly refers to the Component Object Model (COM). COM is a binary-interface standard for software components introduced by Microsoft. It enables communication between software components, allowing them to interact regardless of the programming language they were written in. In C#, you might encounter the term "COM" when working with interop to communicate with COM-based components or APIs.

**\*\*Xamarin\*\*** is a cross-platform app development framework that allows developers to create native mobile applications for iOS, Android, and Windows using a single codebase. Acquired by Microsoft, Xamarin utilizes the C# programming language and the .NET framework, enabling developers to share a significant portion of code across different platforms.

**\*\*.NET Core\*\*** is an open-source, cross-platform framework developed by Microsoft. It serves as a modular and lightweight version of the traditional .NET Framework. .NET Core is designed to be cross-platform, supporting Windows, Linux, and macOS, making it versatile for a variety of applications.

**\*\*Unity C#\*\*** refers to the programming language C# (C Sharp) within the context of the Unity game development engine. Unity is a popular cross-platform game development framework that allows developers to create interactive and immersive experiences for various platforms, including PC, consoles, mobile devices, and augmented reality/virtual reality (AR/VR) devices.

**C#** is the primary scripting language used in Unity. Game developers write C# scripts to define the behavior and functionality of game objects, control the game logic, handle user input, and implement various features. Unity's use of C# provides a powerful and flexible environment for game development.

**\*\*REST\*\*** stands for Representational State Transfer, and it is an architectural style for designing networked applications. It defines a set of constraints that, when applied to web services, create a scalable and stateless communication system. RESTful web services use standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources identified by URLs.

C. Critically,explain ANY three key functions of CLR(50 words):

**\*\*Just-In-Time Compilation (JIT)\*\*** is a process in the Common Language Runtime (CLR) of the .NET framework where code written in Intermediate Language (IL) is dynamically compiled into native machine code during runtime. This on-the-fly compilation ensures that the code is optimized for the specific execution , performance and allowing independence at the IL level.

**\*\*Reduced Footprint:\*\*** Since the JIT compiler generates native code only for the parts of the application that are actually executed, it helps reduce the memory footprint by not compiling the entire program upfront.

**\*\*Adaptability:\*\*** JIT compilation enables .NET applications to be platform-independent at the IL level while still benefiting from native performance optimizations on each specific platform.