**Name: FATOYE BABATUNDE DAYO**

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

The .NET Framework was first released in 2002 and has gone through numerous relevant variants since then, including .NET Framework 2.0, 3.0, 3.5, 4.0, 4.5, 4.6, 4.7, 4.8 and 5.0. Each new version has added new features, improved performance, and addressed stability vulnerabilities. It introduced properties like generics, nullable types, iterators, and anonymous procedures that made C# more expressive and have allowed developers to write cleaner, more efficient code. In 2007, C# 3.0 was released, which introduced two innovative features: language-built query (LINQ) and anonymous types. C# 5.0, released in 2012, introduced the async and await keywords, which allows writing asynchronous code. These updates introduced properties such as non-final named arguments, boundary modifiers, and improved read-only members. It introduced properties like allusion types that allow null values, shift expressions, and asynchronous transmissions. These additions improved code stability, decreased common programming errors, and made it easier to work with asynchronous data streams. In 2016, Microsoft introduced .NET Core, an open source cross-platform framework that allowed developers to build applications that could run on Windows, macOS, and Linux.

* **Explain the following terms** ;

**MONO:**

Mono is an open source implementation of the .NET Framework that runs on a variety of platforms, including Linux, macOS, iOS, Android, and Windows. It was originally developed by Ximian but later acquired by Novell, but it is now maintained by Xamarin(a subsidiary of Microsoft). Mono makes it possible to develop and run .NET applications on platforms that don't have native support for the .NET Framework. It was developed using C, C#, XML and it was released in June 2004.

**XAMARIN:**

Xamarin is a Microsoft-owned software company founded in May 2011 by the engineers that created Mono, Xamarin.Android (formerly Mono for Android) and Xamarin.iOS (formerly MonoTouch), which are cross-platform implementations of the Common Language Infrastructure (CLI) and Microsoft .NET. Xamarin is a cross-platform development framework for mobile applications. It uses C# and the .NET Framework to create native iOS, Android, and Windows applications.

**COM:**

Component Object Model, is a technology that allows software components to communicate with each other, regardless of the language or platform they were written in. .NET applications can use COM to interact with older Windows applications and system components. COM was developed by Microsoft and has been a core part of the Windows operating system since the 1990s.

**.NET Core:**

.NET Core is a new version of the .NET Framework that is designed to be modular and lightweight. It is open source and It is a cross-platform framework that runs on Windows, macOS, and Linux operating systems. .NET Core Framework can be used to build different types of applications such as mobile, desktop, web, cloud, IoT, machine learning, microservices, game, etc. Other features are provided as NuGet packages, which you can add it in your application as needed. In this way, the .NET Core application speed up the performance, reduce the memory footprint and becomes easy to maintain.

**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. Unity can target 25+ platforms across mobile, desktop, console, TV, VR, AR, and the web. It has a large community of developers and assets that make it easy to get started.

**REST:**

Representational State Transfer, is an architectural style for building web services. RESTful web services use HTTP and JSON to send and receive data. The REST architectural style is designed for network-based applications, specifically client-server applications. REST has been employed throughout the software industry to create stateless, reliable web-based applications. An application that adheres to the REST architectural constraints may be informally described as RESTful,

* **Critically, explain ANY three key functions of CLR(50 words)**

\* Automatic memory management: CLR calls various predefined functions of .NET framework to allocate and de-allocate memory of .NET objects. So that, developers need not to write code to explicitly allocate and de-allocate memory.

\* Garbage collection: Garbage collection is the major role of CLR which prevents memory leaks during execution of programs. The garbage collector of CLR automatically determines the best time to free the memory, which is reserved by an object for execution.

\* JIT compilation: JIT stands for Just In Time. It is also an important part of Common Language Runtime (CLR), JIT compiler converts MSIL code to targeted machine code for execution.