NAME:ALATISE AMINAT OMOLABAKE

DEPARTMENT: COMPUTER SCIENCE

LEVEL: HND 1

COURSE CODE: COM 316

C# Assignment

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

1. Critically explain any three key Function of CLR (50 words)

Solutions

The .NET Framework, introduced in 2002, has evolved to become a versatile platform for building various types of applications. C#, its primary language, has advanced with features like async/await, LINQ, and pattern matching. The framework transitioned to .NET Core, offering cross-platform support, and in 2020, it evolved into .NET 5, unifying the platform. The recent iteration, .NET 6, emphasizes performance and developer productivity. C# has gained modern constructs such as records and top-level statements. Both .NET and C# continue to adapt to industry needs, enabling robust and efficient application development across diverse environments.

Explain the following terms

- Mono: Mono is an open-source implementation of the .NET Framework, allowing developers to build and run cross-platform applications. It provides a runtime, a set of class libraries, and development tools, enabling the creation of .NET applications for various operating systems, including Linux, macOS, and Windows.

- Xamarin: Xamarin is a platform for building native mobile applications using C# and .NET. It allows developers to share code across multiple platforms, such as iOS, Android, and Windows, while still providing a fully native user experience. Xamarin.Forms, a part of Xamarin, enables the creation of cross-platform UIs using a single, shared codebase.

- COM: COM (Component Object Model) is a binary-interface standard for software components introduced by Microsoft. It enables interprocess communication and dynamic object creation in a language-independent manner. COM components can be used in various programming languages and are a fundamental technology for building Windows-based software.

- .NET Core: .NET Core is an open-source, cross-platform version of the .NET framework. It is optimized for building modern, cloud-based, and containerized applications. .NET Core supports multiple operating systems and architectures and provides a modular and lightweight runtime, making it suitable for a wide range of development scenarios.

- Unity C#: Unity is a popular game development platform that uses C# as its primary scripting language. C# is used to create gameplay mechanics, implement game logic, and interact with the Unity API to build interactive and immersive gaming experiences.

- REST: REST (Representational State Transfer) is an architectural style for designing networked applications. It relies on a stateless, client-server communication model, and leverages HTTP to access and manipulate resources. RESTful APIs use standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources, providing a scalable and interoperable approach for building web services and APIs.

Critically,explain ANY three key functions of CLR

1. Memory Management: The Common Language Runtime (CLR) provides automatic memory management through its garbage collection mechanism. This feature frees developers from manual memory allocation and deallocation, reducing the risk of memory leaks and dangling pointers while improving the overall stability of the applications.

2. Exception Handling: CLR offers robust support for exception handling, allowing developers to write reliable and maintainable code by providing a structured approach to handling runtime errors. This capability simplifies the identification and resolution of issues, enhancing the overall resilience of applications.

3. Security: CLR enforces various security measures such as code access security, role-based security, and cryptographic services, ensuring that applications running within the CLR environment are protected from unauthorized access, malicious code, and potentially harmful actions. These security functions help in creating a secure execution environment for managed code, safeguarding both the application and the system it runs on.