NAME: IGBOJEKWE VICTORIA NMESOMACHUKWU

MATRIC NO: H/CS/23/1067

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

1. The evolution of .NET Framework and C#

The evolution of the .NET framework and C# is like a journey of continuous improvement and innovation. The .NET Framework and C# were introduced, providing developers with a modern platform and language for building windows applications. C# offered a blend of familiar syntax and modern features, while the .NET Framework provided a rich set of libraries and tools. Over the years both the .NET Framework and C# saw significant expansion and enhancement. New versions of C# introduced features like generic LINQ, ASYNC/AWAIT, and improved language constructs. The .NET Framework evolved with updates, adding support for web development (ASP.NET), cloud computing (AZURE), and more.

1. Explain the following terms

Mono

it is an open-source implementation of the .NET framework allowing developers to build and run .NET applications on various platforms, including Linux, MacOS, and Windows. It was developed by Xamarin, now a part of Microsoft and the mono project community. Mono provides a runtime environment, libraries and development tools compatible with the .NET framework enabling cross-platform development.

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 access to native APIs and performance. Xamarin uses mono for compiling C# code into native binaries, offering a powerful solution for developing cross-platform mobile apps with a single codebase.

COM (Component object model)

COM is a binary-interface standard for software components introduced by Microsoft. It enables inter-process communication and dynamic object creation in a language independent manner. COM components can be used across different programming languages and technologies on windows system.

.NET Core

It is an open-source, cross-platform implementation of the .NET Framework. It was initially developed by Microsoft and released in 2016. .NET Core is optimized for modern Cloud-based applications and supports development on windows, MacOS and Linux. It provides a modular and light weight runtime libraries and development tools allowing developer to build high performance, scalable applications across different platforms. In 2020, .NET Core was unified with Xamarin and Mono into the unified .NET5 platform, providing a single framework for diverse workloads.

Unity C#

Unity is a popular cross platform game engine developed by unity technologies. It allows developers to create interactive 2D, 3D, virtual reality and augmented reality experiences for various platforms including desktop, mobile, consoles and web. C# is the primary programming language used in unity for scripting gameplay, controlling objects, implementing game mechanics and creating custom functionality. C# in unity is used for writing scripts, which are attached to game objects and executed during gameplay, allowing developers to bring their games to life with code.

REST (Representational state transfer)

REST is an architectural style for designing networked applications, commonly used in web services development. It emphasizes a stateless client-server communication model, where client request resources from servers using standard HTTP methods (such as GET, POST, PUT, DELETE) and servers respond with representations of the requested resources e.g. (JSON or XML). RESTful APIs adhere to certain principles, including uniform resource identifiers (URLs) for resources, stateless communication and the use of standard HTTP methods, making them scalable, interoperable and easy to understand and implement.

1. Critically, explain any three key Function of CLR

Garbage collection - CLR manages memory by automatically reclaiming unused objects, preventing memory leaks and optimizing resource usage

Just-in-time compilation (JIT) – it converts intermediate language (IL) code into native machine code at runtime, enhancing performance by dynamically optimizing code execution.

Exception-Handling – CLR provides robust error handling mechanisms ensuring reliable and predictable application behavior by catching and managing exceptions during runtime.