Answer to Question 1

The evolution of the .NET framework reflects Microsoft's commitment to adapt to changing industry trends, embrace open-source practices, and provide developers with a powerful and versatile platform for building a wide range of applications.

Since its emergence in 2002, the trajectory of the .NET Framework has been marked by substantial evolution. Originally tailored for Windows desktop applications, it progressively expanded its reach through the integration of ASP.NET and the introduction of features such as WCF and WPF. The pivotal moment arrived in 2016 with the advent of .NET Core, ushering in a new era of cross-platform development and embracing an open-source paradigm. The subsequent amalgamation of .NET Framework and .NET Core resulted in the unified .NET 5.0 in 2020, streamlining the development landscape. In parallel, C#, the principal language for .NET, underwent its own evolution, incorporating modern features, asynchronous programming, and refined syntax. The current synergy of .NET 6.0 and C# underscores a commitment to performance, adaptability, and contemporary development methodologies.

Answer To Question 2

Mono is a cross-platform runtime environment, enabling C# applications to run on different operating systems e.g Linux and macOS, extending the reach beyond Microsoft's

Xamarin is a framework that allows developers to create cross-platform mobile applications using C#.

COM, or Component Object Model, is a Microsoft technology that enables inter-process communication and object-oriented programming. COM is often used for integrating with legacy systems(outdated systems) or accessing components written in other languages.

NET Core is an open-source, cross-platform framework developed by Microsoft for building modern, scalable, and high-performance applications.

.NET Core provides a runtime, libraries, and development tools that allow you to create applications that can run on Windows, macOS, and Linux.

Unity is a popular game development platform that uses C# as its primary scripting language. Unity allows developers to create 2D, 3D, augmented reality (AR), and virtual reality (VR) applications for various platforms, including desktop, mobile, and consoles

REST(Representational State Transfer) is an architectural style for designing networked applications. In C# and in general programming.

Answer to Question 3

1. Thread Management:Thread management in CLR involves handling and coordinating multiple threads to achieve parallelism. Threads are independent units of execution within a process.

CLR provides features for effective thread management:

Creation and Synchronization: CLR allows developers to create and manage threads easily.

Synchronization mechanisms like locks, mutexes, and semaphores help control access to shared resources, preventing data corruption in multi-threaded environments.

2. Type Checking and Verification: Type checking and verification in CLR ensure the integrity and safety of types used in .NET applications. Here are the key aspects

Type Verification during Just-In-Time Compilation: Before converting Intermediate Language (IL) code into native machine code, CLR performs type verification.

Compile-Time Type Checking:

Runtime Type Checking:

3. Debugging support is a development process which is facilitated by tools provided by the Common Language Runtime (CLR) and integrated development environments (IDEs) like Visual Studio. E.g

Symbolic Information:Debugging relies on symbolic information embedded in the compiled code. When C# code is compiled, additional metadata is generated, including details about variable names, function names, and code structure.