**Question 1: Write a short note on the evolution of .Net Framework and C#(100 words)**

.NET Framework and C# have come a long way since 2002. They've expanded to support cross-platform development with .NET Core and evolved into .NET 5 and later versions. C# has grown to be a versatile language with additions like async/await, LINQ, and pattern matching. Both technologies prioritize performance, cross-platform compatibility, and developer productivity.

**Question 2: Explain the following terms: Mono, Xamarin, COM, .Net Core, Unity C#, REST**

Certainly! Here's an explanation of each term:

1. Mono: An open-source implementation of the .NET Framework for building .NET-based applications on platforms other than Windows.

Xamarin: A platform for building cross-platform mobile apps using C# and .NET framework.

COM: A binary-interface standard for software components that enables software components to interact with one another in a networked environment.

.NET Core: An open-source, cross-platform version of the .NET framework, designed for building modern, cloud-based applications.

Unity: A cross-platform game engine developed by Unity Technologies for creating interactive 2D, 3D, VR, and AR experiences across multiple platforms.

C#: A modern, object-oriented programming language developed by Microsoft for building a variety of applications.

REST: A software architectural style that defines a set of constraints for creating scalable web services based on the principle of a client-server architecture, statelessness, cacheability, layered system, and uniform interface.

**Question 3: Critically, explain ANY three key functions of CLR (50 words)**

1. **Type Safety: CLR ensures type safety by verifying the types and their interactions at both compile and runtime, preventing type-related errors.**
2. **Code Access and Verification: CLR performs code access and verification to ensure that the code accessing resources meets security requirements, maintaining system integrity.**
3. **Execution Engine: CLR includes an execution engine that converts Intermediate Language (IL) code into machine code during runtime, optimizing performance and enabling platform independence.**