**Name: Sofela Goodnews Inioluwa**

**Application number: FPI/HND/2023/00004276**

**Course code: Com 316**

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

The .NET Framework and C# have evolved significantly over the years. The .NET Framework was first introduced by Microsoft in the early 2000s as a platform for building Windows applications. It provided a common runtime environment and a set of libraries for developers to work with. C# was developed as a modern, object-oriented programming language specifically designed for the .NET Framework. Over time, both the .NET Framework and C# have undergone updates and improvements, with the introduction of new features and enhancements to make development easier and more efficient.

**2. Explain the following terms ;**

- **Mono: I**t's like a free version of the .NET Framework that lets you run .NET apps on different operating systems like Linux and macOS.

**- Xamarin:** It's a cool framework that uses C# and lets you build mobile apps for iOS, Android, and Windows using just one codebase.

**- COM:** It's a tech used for creating and accessing software components in a distributed environment.

**- .NET Core:** It's an open-source version of the .NET Framework that works on Windows, Linux, and macOS.

**- Unity: I**t's a popular game development platform that uses C# for scripting.

**- REST: I**t's a style for designing networked apps where resources are represented as URLs and accessed using standard HTTP methods.

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

**Garbage Collection**: CLR’s garbage collector automatically manages the memory allocated to objects, freeing up resources that are no longer in use. It tracks object references, identifies unused objects, and reclaims memory, preventing memory leaks and improving application performance.

**b) Just-in-Time (JIT) Compilation:** CLR’s JIT compiler translates Intermediate Language (IL) code into native machine code at runtime. This compilation process optimizes performance by converting IL code into machine-specific instructions, making it more efficient for the underlying hardware to execute.

**c) Exception Handling**: CLR provides a robust and structured exception handling mechanism. It allows developers to catch and handle exceptions, ensuring that the application gracefully handles runtime errors. CLR’s exception handling promotes code reliability, maintainability, and fault tolerance by separating error handling logic from the main program flow.