**Questions for this assignment**

What is an assembly in C#?

What are fields in C#?

Explain all access modifiers in C#?

What is the difference between public and private access modifiers in C#?

What is the default access modifier for a class member in C#?

What is the default access modifier for a class (or other types) in C#?

What is the purpose of using access modifiers in C#?

Difference between 'internal' and 'protected internal' in C#

Difference between 'protected' and 'private protected' in C#

What is a constant in C#?

Can a constant be modified after it is declared?

What is the difference between a constant and a readonly variable in C#?

What is an assembly in C#?

In C#, an assembly is a logical unit of code that consists of one or more files, typically in the form of executable code (such as .exe files) or dynamically linked libraries (DLLs) that can be loaded and executed by the Common Language Runtime (CLR). An assembly is the fundamental building block of .NET applications and is the primary mechanism for packaging, deploying, and versioning code in .NET.

An assembly can contain types (such as classes, structures, and interfaces), resources (such as images, icons, and strings), metadata (such as version information and security permissions), and code (such as IL or intermediate language instructions). The code within an assembly is compiled into a format that is understandable by the CLR and can be executed on any platform that supports the CLR.

What are fields in C#?

Fields in C# are variables that hold data within a class or struct. They are used to store data that is specific to an object, and can be accessed and modified through object instances or directly through the class or struct itself.

Explain all access modifiers in C#?

Access modifiers in C# determine the level of access that a class member has. There are five access modifiers in C#: public, private, protected, private protected, protected internal and internal.

What is the difference between public and private access modifiers in C#?

Public members can be accessed from anywhere, both within and outside the class, whereas private members can only be accessed within the class they are defined in.

What is the default access modifier for a class member in C#?

The default access modifier for a class member in C# is ‘private’.

What is the default access modifier for a class (or other types) in C#?

The default access modifier for a class / other types in C# is ‘internal’.

What is the purpose of using access modifiers in C#?

Access modifiers help in encapsulating and hiding the internal workings of a class or struct, and controlling the level of access to its members.

Difference between 'internal' and 'protected internal' in C#

'internal' is an access modifier that allows members to be accessed from any code within the same assembly, but not from code outside the assembly. This means that all classes in the same assembly can access internal members, but classes in other assemblies cannot.

'protected internal' is an access modifier that allows members to be accessed from any code within the same assembly or from derived classes outside the assembly. This means that all classes in the same assembly, as well as any derived classes in other assemblies, can access protected internal members.

In other words, 'internal' grants access to members only within the same assembly, while 'protected internal' grants access to members within the same assembly as well as to derived classes outside the assembly.

Difference between 'protected' and 'private protected' in C#

'protected' is an access modifier that allows members to be accessed from within the class and its derived classes, whether they are in the same assembly or a different assembly. This means that any derived class, whether it is in the same assembly or a different assembly, can access the protected member.

'private protected' is an access modifier that allows members to be accessed from within the class and its derived classes that are in the same assembly, but not from derived classes in a different assembly. This means that only derived classes that are defined in the same assembly as the base class can access the private protected member.

In other words, 'protected' grants access to members from all derived classes, while 'private protected' limits access to derived classes in the same assembly only.

What is a constant in C#?

A constant in C# is a variable whose value cannot be changed once it is assigned. It is declared using the const keyword, and its value must be initialized at the time of declaration.

Can a constant be modified after it is declared?

No, a constant cannot be modified after it is declared.

What is the difference between a constant and a readonly variable in C#?

A constant is a **compile-time constant** whose value cannot be changed, whereas a readonly variable is a **runtime constant** whose value can only be set once, either at initialization or in a constructor.