C# (Some Important Terms)

INSTRUCTOR: ADEEN BIN WAHEED

LECTURE 9

Important Notes

- Always declare class methods and constructors to be public.
- In case of Interface, do not specify the modifier to be public.

Static Classes

A static class is basically the same as a non-static class, but there is one difference: a static class cannot be instantiated. In other words, you cannot create a object of the static class. Static Classes must contain static member variables and static member functions. Because there is no instance object, you access the members of a static class by using the class name itself.

For Example I have a static class named A having a method Fun() than I will call this method as:

A.Fun();

Static Variables

In C#, we don’t have global variables but still we can use the static variables as global variables because whatever we will assign value to static variable than it will remain the same for the whole program.

Syntax:

```
public static int id=1;
```

Static Classes or Static Variables

Running Demo

Abstract Classes

The abstract keyword enables you to create classes and class members that are incomplete and must be implemented in a derived class.

For a abstract class it is compulsory to declare one or more than one of the its method to be abstract method.

Other its functionality is same like Interface.

Abstract Classes

Running Demo

Sealed Classes

The sealed keyword enables you to **prevent the inheritance of a class** or certain class members that were previously marked virtual. Means a sealed class cannot be used as a base class.

All subclasses of a sealed class (whether it's abstract or not) must be in the same file as the sealed class.

Sealed Classes

Running Demo