**Object Oriented Programming Langage Part 2**

**8. What are different access modifiers in .Net?**

* **Private** - The type or member can only be accessed by code in the same class or struct.
* **Protected** - The type or member can only be accessed by code in the same class or struct, or in a derived class.
* **Internal** - The type or member can be accessed by any code in the same assembly, but not from another assembly.
* **Procted Internal**- The type or member can be accessed by any code in the same assembly, or by any derived class in another assembly.
* **Public** -The type or member can be accessed by any other code in the same assembly or another assembly that references it.

**Note:** In VB.Net 'Internal' is called as 'Friend'

**11. What is mean by Partial method?**

Partial methods are methods defined in a partial class that are (optionally) divided across two files. With partial methods one file contains the method signature - the method name, its return type, and its input parameters - while the body is (optionally) defined in a separate file. If the partial method's body is not defined then the compiler automatically removes the partial method signature and all calls to the method at compile-time.

Example for Partial method

{

string m\_Name;

public String Name

{

get { return m\_Name; }

set { m\_Name = value; }

}

public partial string GetEmpDetails(string ID);

}

partial class Employee

{

int m\_Age;

public int Age

{

get { return m\_Age; }

set { m\_Age = value; }

}

public partial string GetEmpDetails(string ID)

{

return "Employee1";

}

}

**12. Why do we go for Partial method?**

Partial methods are mainly useful in auto-generated code situations. A code generating tool might know that there are certain extension points that some users are going to be interested in customizing. For example, the objects created in LINQ to SQL have partial methods like OnLoaded, OnCreated, OnPropertyNameChanging, and OnPropertyNameChanged. The auto-generated code calls the OnCreated partial method from its constructor. If you want to run custom code when one of these objects is created you can create a partial class and define the body for the OnCreated partial method.

**13. Why do we go for Partial class?**

1. Improve the readability of extremely large classes by partitioning related methods into separate files.
2. Partial classes enable the code generator to generate code in one file while the developer who may need to extend the auto-generated logic can do so in a separate file, which eliminates the worry that the code generator might overwrite a developer's customizations.

**14. Where we use Partial method and class?**

Partial classes and partial methods are most commonly used in auto-generated code. It provides a simple and safe way to add new functionality or extend existing functionality of auto-generated code.

**15. What are restrictions for Partial method?**

1. Partial definitions must preceded with the key word "Partial"
2. Method signature should be same in both partial class file
3. We cannot have partial implementation in one file and another implementation in other file. We can have declaration in one file and implementation in another file.

**16. What is mean by Static class?**

Static class is used to create attributes and methods that can be accessed without creating the instance of the class. Static classes are loaded automatically by the .NET Framework when application or assembly is loaded. *'Static'* key word is used to mention the static class. e.g MyStaticClass.PI

Example for Static Class

public static class MyStaticClass

{

public static decimal PI = 3.14M;

public static int Add(int num1, int num2)

{

return num1 + num2;

}

public static string Append(string str1, string str2)

{

return str1 + str2;

}

}

MyStaticClass.PI

**17. What is mean by Static method?**

Static method can be accessed without creating the instance of the class. 'Static' key word is used to mention the static method. Static methods can be created inside the normal class or static class. If we create the static method inside the normal class, static method will not be able to access by creating instance of the class. e.g Math.Add()

**18. Can we override static method?**

No, compiler will not allow overriding the static method.

**19. What are uses of static class and method?**

1. Compiler will not allow creating the instance of the class
2. Static class also makes the implementation simpler and faster
3. Cannot inherit a static class since it is sealed

**20. What is static constructor?**

A static constructor is used to initialize any static data, or to perform a particular action that needs performed once only. It is called automatically before the first instance is created or any static members are referenced.

**Example:**

public class MyStaticClass

{

static int count;

static MyStaticClass()

{

count = 0;

Console.WriteLine("Static class is initialized");

}

public static void MyMethod(string name)

{

Console.WriteLine("Static class is initialized " + name);

}

}

MyStaticClass.MyMethod("John");

**Output:**

Static class is initialized

Hello John