Partial Classes

Partial class is a new concept in .net 2005. A partial class is a class whose definition is divided in to more than one file. For example in windows application a windows form is divided in to two files, a file with extension **.cs** and a file with extension **.designer.cs.** File with extension **.cs** contains the code written by the programmer and file with extension **.designer.cs** contains automatically generated code for the design by visual studio.net. This is possible only with partial classes.

In the same way in ASP.net, a web page is divided in to two files, a file with extension **.Aspx** and a file with extension **.Aspx.cs**. File with extension **.aspx** contains the design related code of the web page and file with extension **.Aspx.cs** contains the code written by the programmer for the web page. This is also possible only with partial classes.

In this way when you have to divide a class into more than one file, then use partial classes. To create a class as partial class, use the keyword **partial** while creating the class. While creating the class as partial class, consider the following points.

1. In every file class must be created with same name.
2. In every file class must be created with the keyword **partial.**
3. In every file class must be created with same access modifier.

Example : The following example creates a console application in C# that contains two files program.cs and class1.cs where a class Test is divided in to these two files.

Program.Cs

```
namespace PartialClasses
{
    partial class Test
    {
        public void M1()
        {
            Console.WriteLine("M1 From Program.Cs");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Test T = new Test();
            T.M1();
            T.M2();
        }
    }
}
```

Class1.Cs

```
namespace PartialClasses
{
    partial class Test
    {
        public void M2()
        {
            Console.WriteLine("M2 From Class1.Cs");
        }
    }
    class Class1
    {
    }
}
```

Nullable Types

Nullable type is another new concept in .net 2005. A nullable type is a type that can store null value along with the valid range of values for its data type. A variable in .net can store only the valid range of values for its data type and it can’t store null value. But when you are working with databases like SQL server or Oracle and the data retrieved from table is stored in to variables then if retrieved data contains null value then an error will be raised. A solution for this problem is nullable types. For declaring a variable as nullable type, suffix the data type in declaration with “**?**”.

DataType? Var;

The following example declares the variable **B** as nullable type of byte.

Byte? B;

When a variable is declared as nullable type, then that variable will have a property called **HasValue**. This property can be used to determine whether nullable type contains a value or null. If this property contains the true then the variable contains a value and when it contains false then the variable contains null.

Example : The following example creates a variable B as nullable type of byte.

```
namespace NullableTypes
{
    class Program
    {
        static void Main(string[] args)
        {
            byte? B;
            B = null;
            if (B.HasValue)
                Console.WriteLine("Value Of B Is {0}", B);
            else
                Console.WriteLine("B Has NULL");
        }
    }
}
```