AGENDA

- Recap
- Object initializer
- Collection initializer
- Nullable types
- Indexer
- Introduction to GUI programming
- Windows Forms
- Windows Forms Components
- Controls used in Windows Forms
- Events

Object initializer in C#

In object initializer, you can initialize the value to the fields or properties of a class at the time of creating an object without calling a constructor. In this syntax, you can create an object and then this syntax initializes the freshly created object with its properties, to the variable in the assignment. It can also place indexers, to initializing fields and properties, this feature is introduced in C# 6.0.

Collection Initializer in C#

Collection initializer is also similar to object initializers. The collections are initialized similarly like objects are initialized using an object initializer. Or in other words, generally, we used the **Add()** method to add elements in collections, but using a collection initializer you can add elements without using Add() method.

Nullable types in C#

In C#, the compiler does not allow you to assign a null value to a variable. So, **C# 2.0** provides a special feature to assign a null value to a variable that is known as the Nullable type. The Nullable type allows you to assign a null value to a variable. Nullable types introduced in C#2.0 can only work with Value Type, not with Reference Type.

The nullable types for Reference Type is introduced later in C# 8.0 in 2019 so that we can explicitly define if a reference type can or can not hold a null value. This helped us to tackle the issue of NullReferenceException without using conditionals

The Nullable type is an instance of *System.Nullable<T> struct*

Indexer in C#

An indexer allows an instance of a class or struct to be indexed as an array. If the user will define an indexer for a class, then the class will behave like a virtual array. Array access operator i.e (**[ ]**) is used to access the instance of the class which uses an indexer. A user can retrieve or set the indexed value without pointing an instance or a type member. Indexers are almost similar to the **Properties**. *The main difference between Indexers and Properties* is that the accessors of the Indexers will take parameters.

Readonly variable

In C#, you can use a ***readonly*** keyword to declare a readonly variable. This readonly keyword shows that you can assign the variable only when you declare a variable or in a constructor of the same class in which it is declared.

Windows Forms

Windows Forms is a Graphical User Interface(GUI) class library which is bundled in *.Net Framework*. Its main purpose is to provide an easier interface to develop the applications for desktop, tablet, PCs. It is also termed as the **WinForms**. The applications which are developed by using Windows Forms or WinForms are known as the **Windows Forms Applications** that runs on the desktop computer. WinForms can be used only to develop the Windows Forms Applications not web applications. WinForms applications can contain the different type of controls like labels, list boxes, tooltip etc.

Windows Forms Controls

Label

Text Box

Radio Button

Check Box

Combo Box

List Box

Tree View

Button

Panel

DateTimePicker

Event

An event is an action on an object.

Example:

button_ click()

Form_load()