1. What is LINQ Union Method?

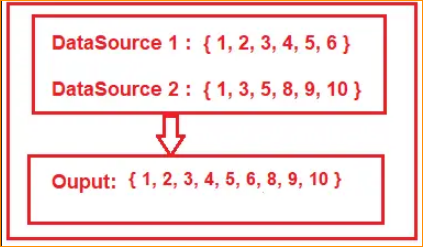
The LINQ Union Method in C# is used to combine multiple data sources into one data source by removing duplicate elements. There are two overloaded versions available for the LINQ Union Method as shown below.



The one and only difference between the above two LINQ Union methods is that the second overloaded version takes IEqualityComparer as an argument. That means when we are working with Complex Data Types, in order to work as expected, we can use the overloaded method which takes the IEqualityComparer parameter.

Examples to Understand LINQ Union Method with Value Type in C#

Let us understand LINQ Union Method with an example. Please have a look at the below image. As you can see in the below image, here we have two integer data sources i.e. DataSource 1 and Data Source 2. DataSource 1 contains elements such as 1, 2, 3, 4, 5, and 6, and DataSource 2 contains elements such as 1, 3, 5, 8, 9, and 10. If we want to retrieve all the elements from both collections by removing the duplicate element i.e. if we want the result as 1, 2, 3, 4, 5, 6, 8, 9, and 10, then we need to use the LINQ Union method.



LINQ Union Method Example using Method and Query Syntax:

The following example shows the use of the LINQ Union Method using both Method and Query Syntax to fetch all elements from both collections by removing the duplicate elements. In query syntax, there is no such operator called Union, so here we need to use the mixed syntax i.e. both the query and method syntax to achieve the same.



1. LINQ Union Method Example with String Array using C#:

Let us see an example to understand how LINQ Union Method works with string collection in C#. Here, in the below example, we have two arrays of countries that also contain the same country name and our requirement is to return all countries from both collections by removing the duplicate country names.





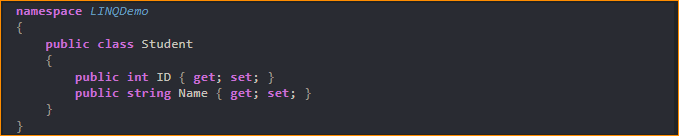
As you can see it displays the country UK twice. This is because the default comparer that is being used by the LINQ Union method is case-insensitive. So if you want to ignore the case-sensitive then you need to use the other overloaded version of the Union method which takes IEqualityComparer as an argument. Let us modify the Main method of the program class as follows to use the StringComparer as an argument that implements the IEqualityComparer interface.





1. LINQ Union Method with Complex Type in C#:

The LINQ Union() Method like other Set Methods (such as Distinct, Expect, and Intersect) also works in a different manner when working with complex data types such as Product, Employee, Student, etc. Let us understand this with an example. Create a class file with the name Student.cs and then copy and paste the following code into it.







1. Using Anonymous Type with Union Method in C#:

In this approach, we need to select all the individual properties to an anonymous type using the LINQ select operator or Select Extension Method. The following program does exactly the same thing.

