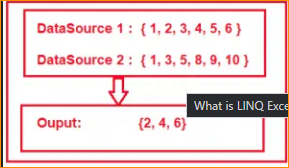
1. What is LINQ Except Method in C#?

The LINQ Except Method in C# is used to return the elements which are present in the first data source but not in the second data source. There are two overloaded versions available for the LINQ Except Method in C#. They are as follows.



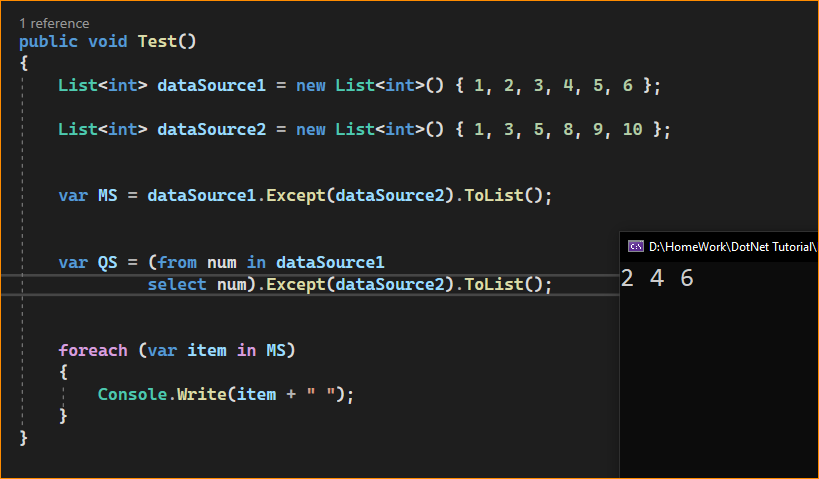
The one and only difference between the above two methods is that the second overloaded version takes IEqualityComparer as an argument. That means the Except Method can also be used with Comparer. Let us understand the LINQ Except Method with an example. Please have a look at the following image.

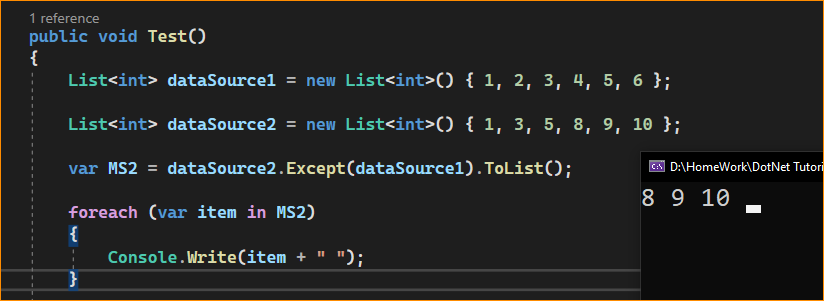


As you can see in the above image, we have two data sources i.e. DataSource 1 and Data Source 2. DataSource 1 contains elements such as 1, 2, 3, 4, 5, and 6, and DataSource 2 contains elements such as 1, 3, 5, 8, 9, and 10. If we want to retrieve the elements such as 2, 4, and 6 from the first data source which does not exist in the second data source, then we need to apply the Except operation between DataSource 1 and Data Source 2. Let us see how to do this with both Query and Method Syntax using Except Method.

1. Example to Understand Except Method with Value Types in C#:

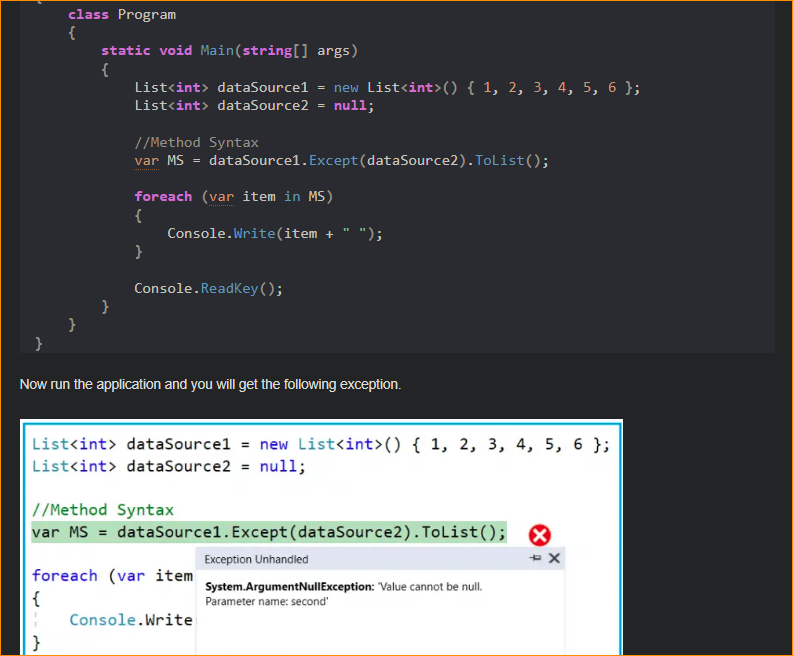
In the below example, I am showing how to use LINQ Except Method with Value Types using both Method and Query Syntax. In query syntax, there is no such operator call Except, so here we have to use both query and method syntax to achieve the same.





1. What happens if any of the sequences is null?

The Except Method will throw an exception if any of the sequences is null. In the below example, the second sequence is null and while performing the Except operation using the Except Method it will throw an exception.

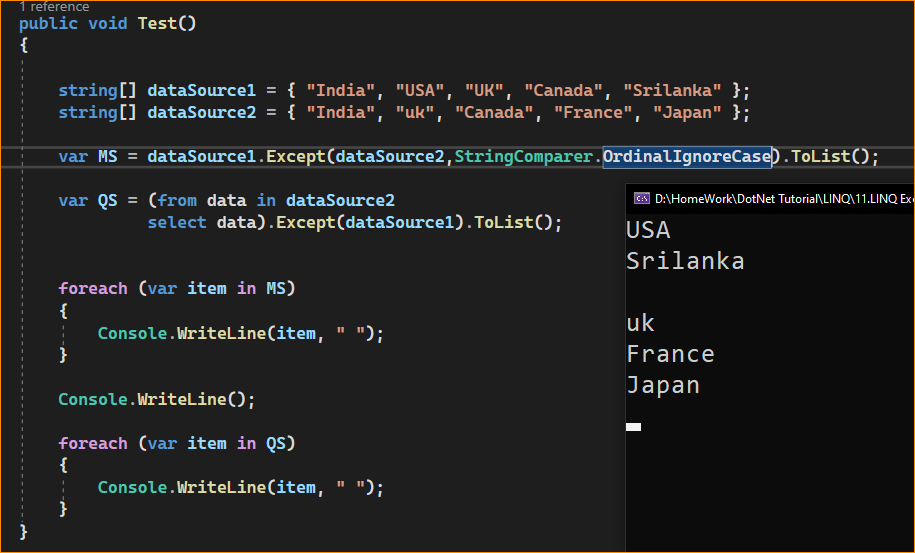


1. Example to Understand LINQ Except Method with String Data Type:

In the below example, I am showing how to use LINQ Except Method with String Data using both Method and Query Syntax. Here, we have a string array of countries and we need to return the countries from the first collection, that are not present in the second collection using Except Method. So, modify the Main Method of the Program class as follows.

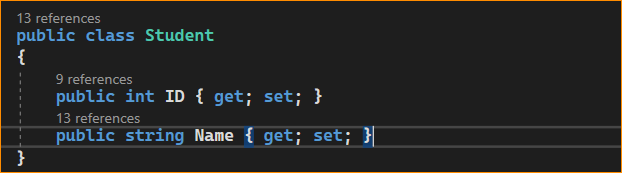


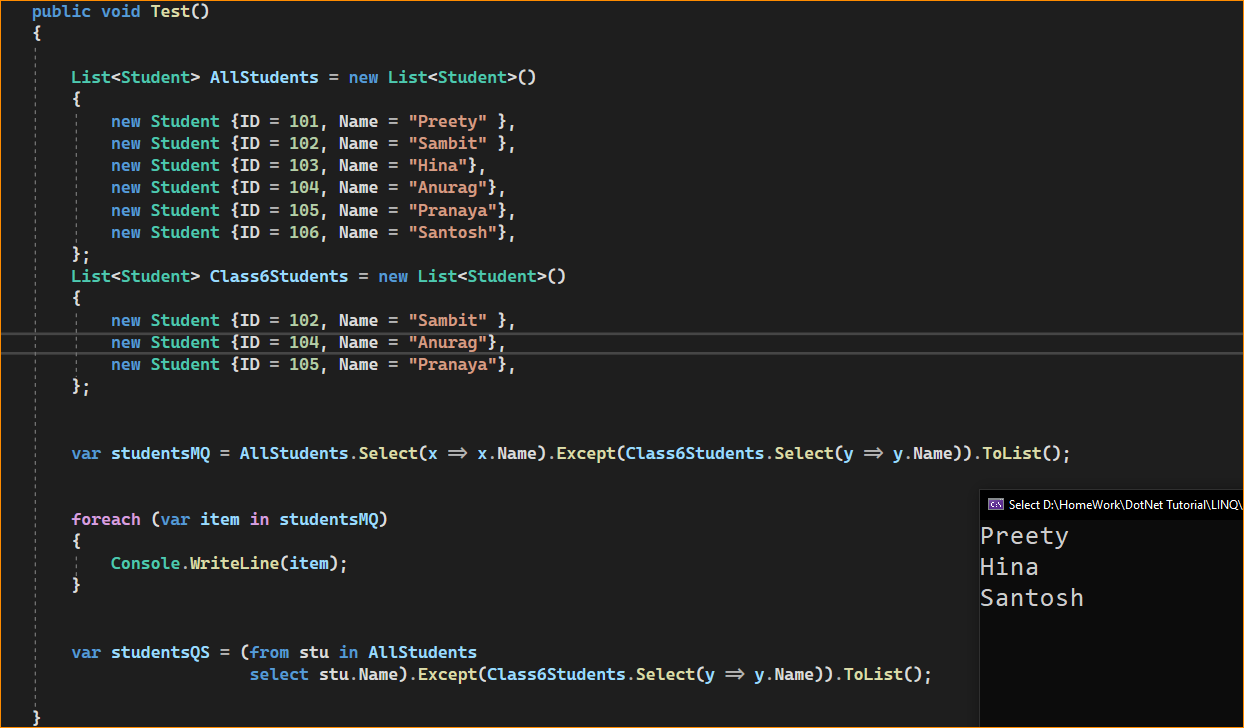
In spite of having the country UK in the second collection, it still shows in the output. This is because the default comparer that is being used to filter the data by the Except Method is case-insensitive. So if you want to ignore the case-sensitive, then you need to use the other overloaded version of the Except() method which takes IEqualityComparer as an argument. So, modify the Main Method of the Program as shown below. Here, we are passing StringComparer as an argument and this StringComparer class already implements the IEqualityComparer interface.



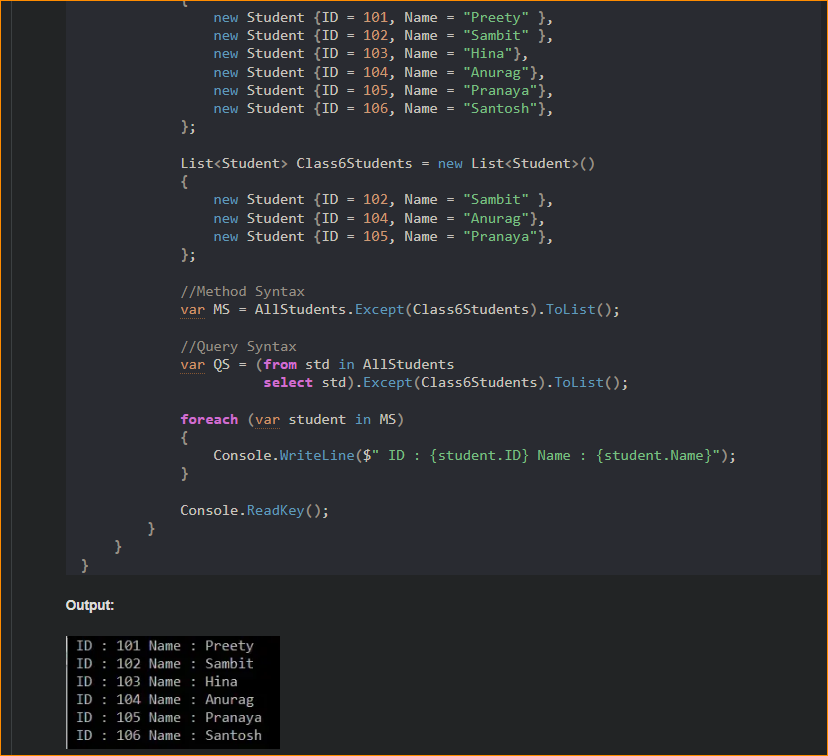
1. LINQ Except() Method with Complex Type in C#:

The LINQ Except() Method in C# works slightly different manner when working with complex types such as Employee, Product, 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.





Now we need to select all the information of all the students from the first data source which is not present in the second data source. Let us modify the Main Method of the program class as shown below.



* Now we need to select all the information of all the students from the first data source which is not present in the second data source. Let us modify the Main Method of the program class as shown below.

Using Anonymous Type with Except Method in C#:

In this approach, we need to select all the individual properties to an anonymous type. The following program does exactly the same thing.

