**LINQ Intersect**

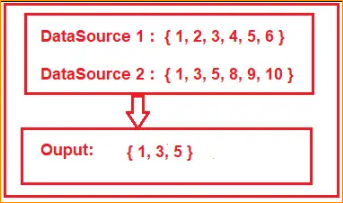
The LINQ Intersect Method in C# is used to return the common elements from both collections. The elements that are present in both data sources are going to be returned by Intersect Method. There are two overloaded versions available for the Intersect Method in C#. They are as follows.



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

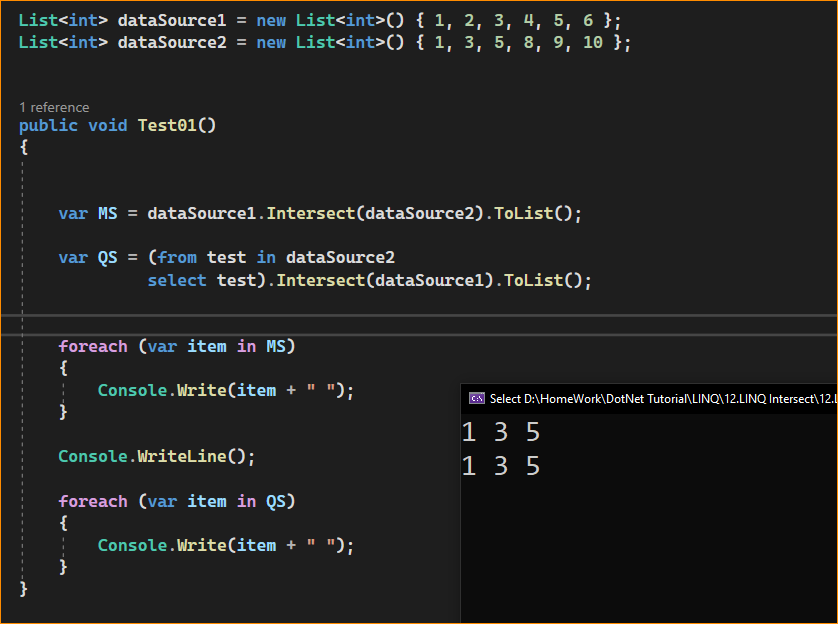
Examples to Understand LINQ Intersect Method with Value Type

Let us understand LINQ Intersect 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 the elements such as 1, 3, and 5 which exist in both the data sources then we need to use the LINQ Intersect method.



LINQ Intersect() Method Example using Method and Query Syntax:

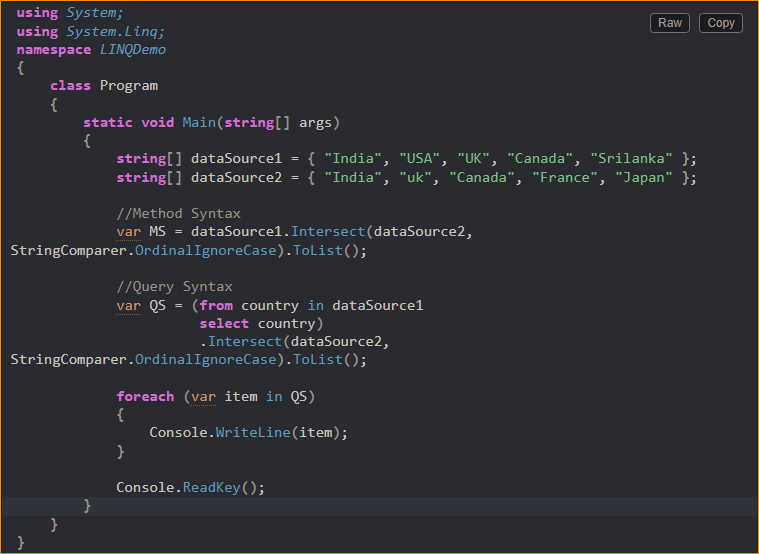
The following example shows the use of the LINQ Intersect() Method using both Method and Query Syntax to fetch the common elements that exist in both collections. In query syntax, there is no such operator called Intersect, so here we need to use the mixed syntax i.e. both the query and method syntax to achieve the same.



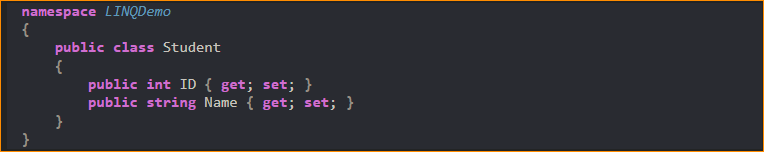
Using string-

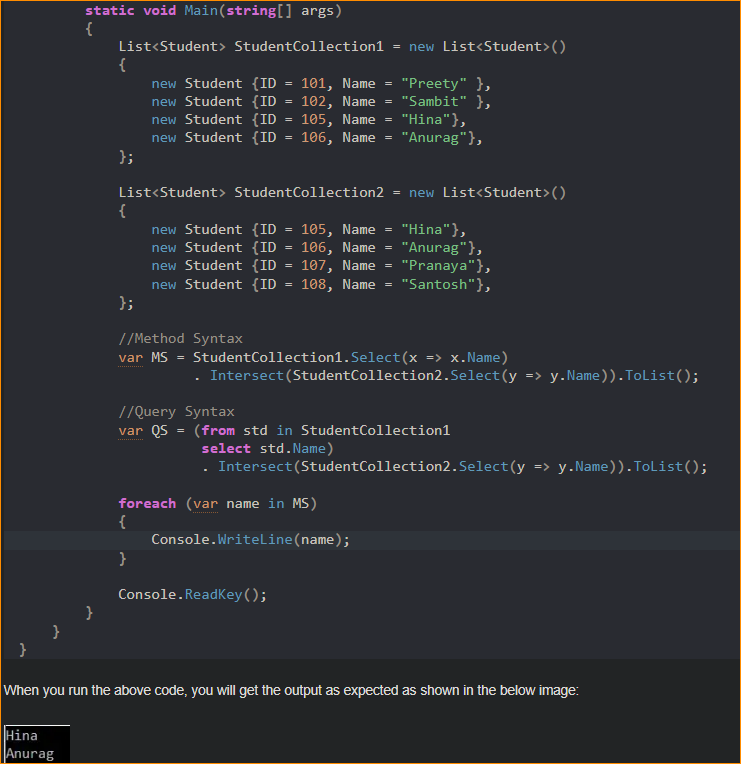


Ignore Case Sensitive –



LINQ Intersect Method with Complex Type in C#:





* Using Anonymous Type with Intersect 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.

