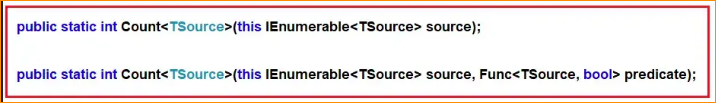
What is LINQ Count Method in C#?

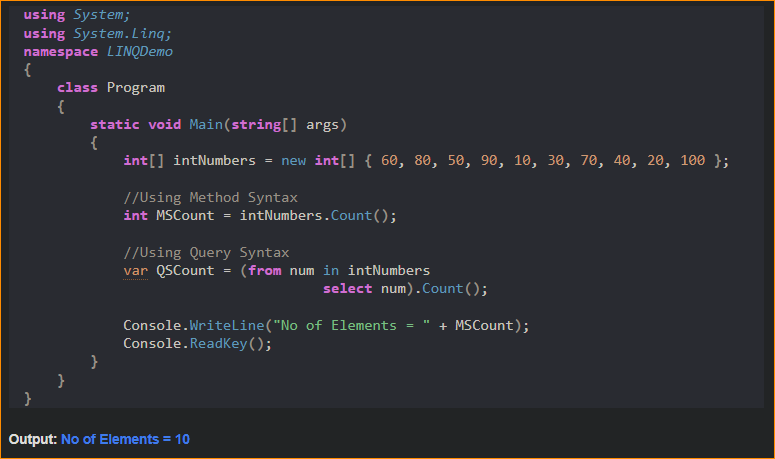
The LINQ Count() Method belongs to the category of Aggregate Operators. The LINQ Count Method is used to return the number of elements present in the collection or the number of elements that have satisfied a given condition. There are two overloaded versions of the LINQ Count() Extension method available. They are as follows.



As you can see in the above image, the first overloaded version does not take any parameter and it simply returns the count of the number of elements present in the collections. On the other hand, the second overloaded version takes one predicate as a parameter and then it returns the count of the number of elements that have satisfied the given condition which we can specify using a lambda expression or by using a predicate function. As you can see in the above image, both method’s return type is int. So, it is always going to return an integer number which indicates the number of elements present in the collection or in the data source.

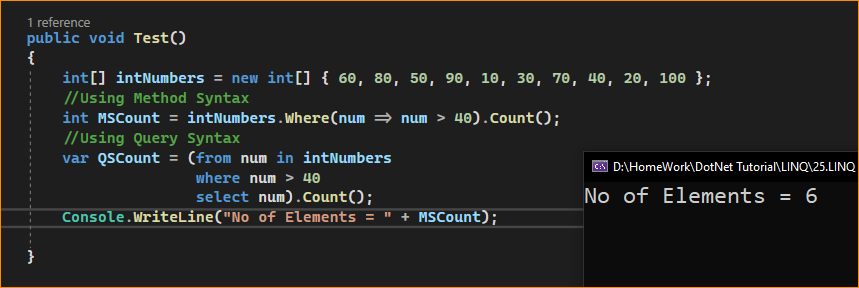
1. Example to Understand LINQ Count Method in C#:

Let us understand the Count() method in C# with Examples. The following example returns the number of elements present in the collection using both Query and Method Syntax. Unlike the Sum, Average, Min, and Max Aggregate functions, we can use Count Method with any data type. In the below example, we are using integer collection. We don’t have any operator called count in the LINQ query syntax, so, we need to use mixed syntax.



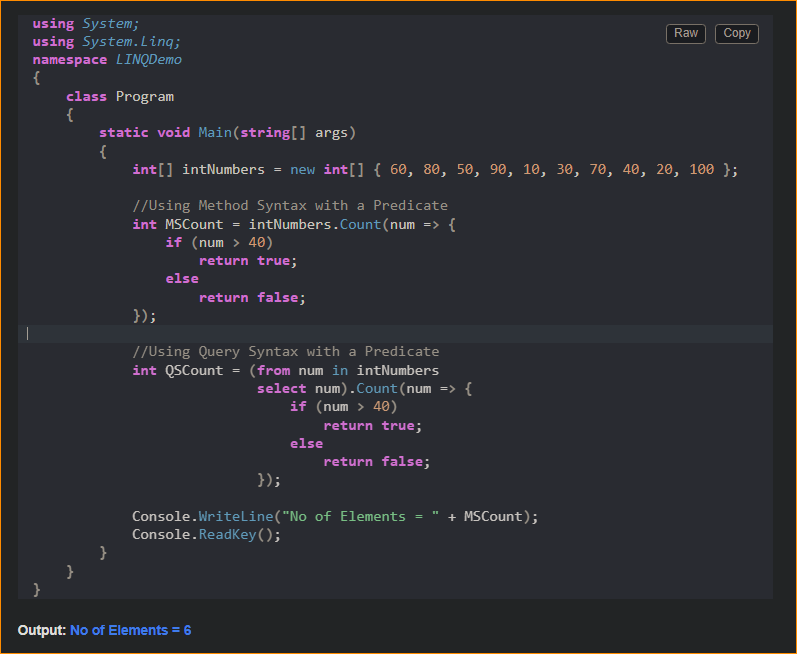
1. Example to Understand Count Method with Where Extension Method in C#

Let us see an example to Understand how we can use the LINQ Count Method along with the LINQ Where Extension Method using both Method and Query Syntax. Now our requirement is to return the number of elements present in the collection which are greater than 40. The following example code exactly does the same.



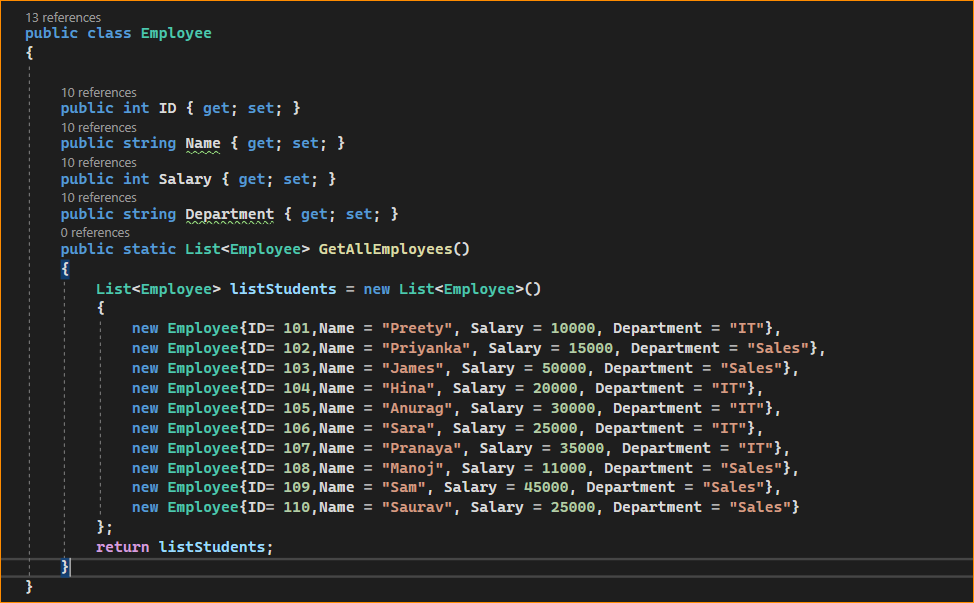
1. Example to Understand How to use LINQ Count Method with Predicate in C#

Let us see an example to Understand How to use LINQ Count Method with Predicate in C# using both Method and Query Syntax. Instead of using the Where Extension method to filter the data, we can also use the other overloaded version of the Count method which takes a Predicate as a parameter, and within that predicate, we can write the logic to filter the data. In the below example, within the Count Method, we are using a Predicate and we are providing the condition whether the number is greater than 40 or not. If the number is greater than 40, then we are returning true else we are returning false. The following example will return the number of elements present in the collection which are greater than 40.

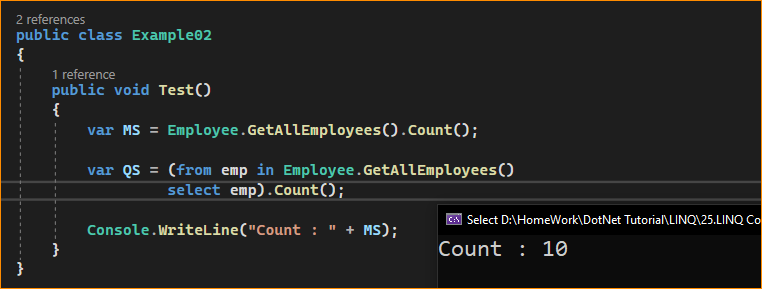


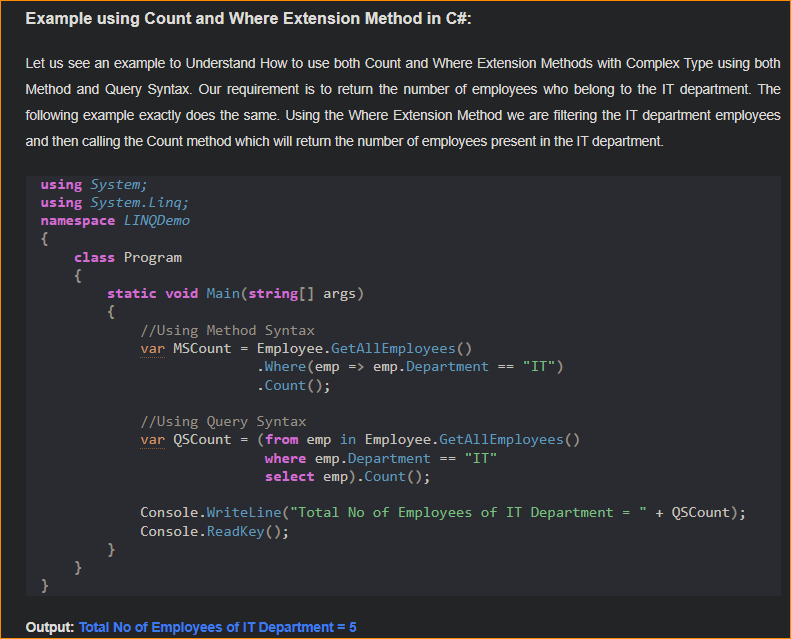
1. Example to Understand LINQ Count Method with Complex Type in C#:

Let us see an example to Understand How to use Count Method with Complex Type in C#. We are going to work with the following Employee class. As you can see, the following Employee class contains four properties i.e. ID, Name, Salary, and Department as well as contains one method i.e. GetAllEmployees() which will return the list of all the employees and this is going to be our data source.



* our requirement is to calculate the Number of Employees present in the collection. The following example returns the Number of Employees present in the collection using the LINQ Count Method.





Let’s rewrite the previous example using the custom predicate. As you can see, within the Count method, we are checking the Department property value and if the Department is It, we are returning true else we are returning false.

