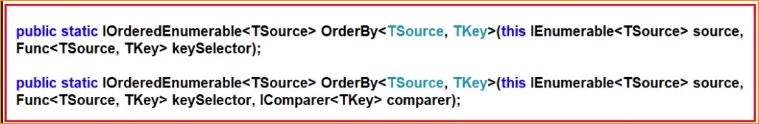
1. What is Linq OrderBy Method?

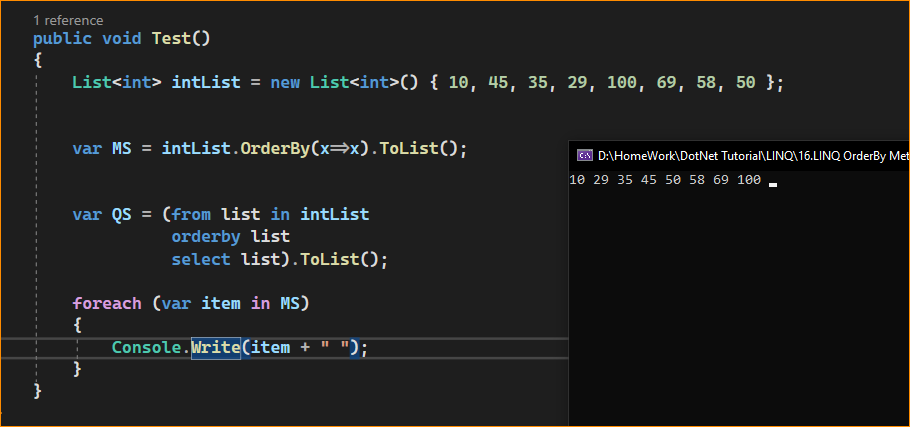
The Linq OrderBy method in C# is used to sort the data in Ascending Order. The most important point that you need to keep in mind is that this method is not going to change the data rather it is just going to change the order of the data. You can use the OrderBy method on any data type i.e. you can use character, string, decimal, integer, etc. There are two overloaded versions of this OrderBy Method available inside LINQ. They are as follows.



The one and only difference between these two overloaded versions are that the second overloaded version takes the IComparer parameter which basically compares the keys when we are creating our custom comparer. Let us understand the use of the LINQ OrderBy method in C# using both query syntax and method syntax.

1. LINQ OrderBy Method with Value Data Type in C#

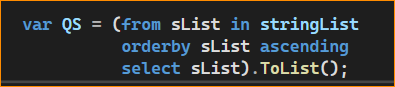
Let us see an example to understand how the LINQ OrderBy Method works with Value Data Type using C#. In the below example, we have a collection of integer data. And then we sort the data in ascending order using the LINQ OrderBy method using both Method and Query Syntax.



1. LINQ OrderBy Method with String Data Type in C#

Let us see an example to understand how the LINQ OrderBy Method works with String Data Type using C#. In the below example, we have a collection of string names. We then sort the data in ascending order using the LINQ OrderBy method with both Method and Query syntax.

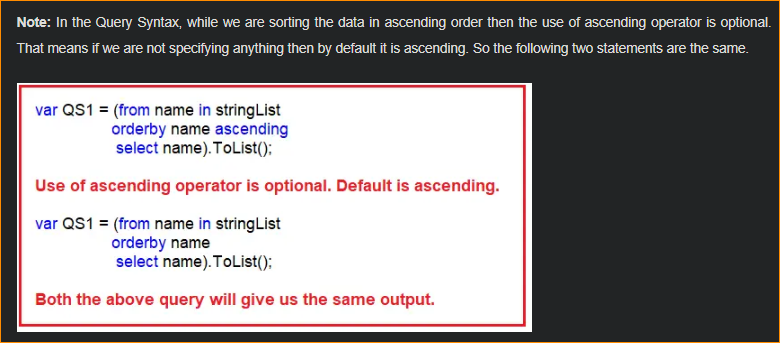




Ascending means to arrange items in an increasing or rising order. When you sort something in ascending order, you are arranging the items from the smallest to the largest, or in alphabetical order from A to Z, depending on the context.

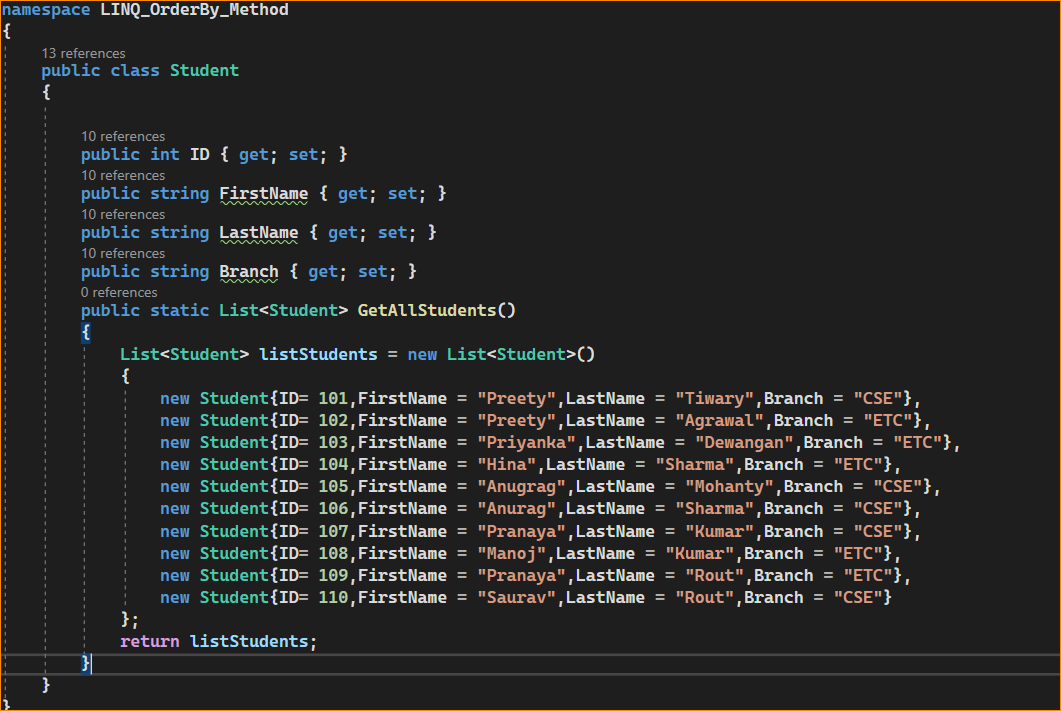
For example, if you have a list of numbers, sorting them in ascending order means that you arrange the numbers from the smallest (lowest) number to the largest (highest) number. If you have a list of words, sorting them in ascending order means arranging them alphabetically from A to Z.

The opposite of ascending order is descending order, which involves arranging items in decreasing or falling order, with the largest or highest values coming first.



LINQ OrderBy Method with Complex Data Type in C#:

Let us see how the LINQ OrderBy Method works with Complex Data Types in C# with some Examples. We are going to work with the following Student class. So, create a class file with the name Student.cs and then copy and paste the following code into it. As you can see, we created the Student class with four properties such as ID, FirstName, LastName, and Brach. We then created one method i.e. GetAllStudents() which is going to return a list of students.

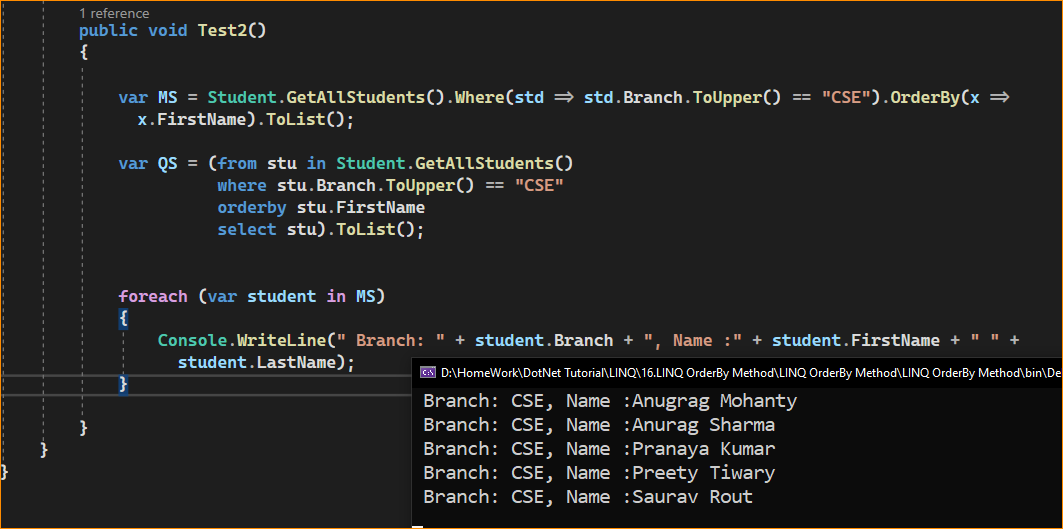


Now, we want to sort the student data based on the Branch of the Student in ascending order. To do so, modify the Main method of the Program class as follows. In the below example, we are Sorting the data in ascending order based on the student branch using LINQ OrderBy Method with both Method and Query Syntax.



1. Sorting with Filtering using LINQ

Now we need to fetch only the CSE branch students and then we need to sort the data based on the FirstName in Ascending order. The most important point that you need to remember is, you need to use the Where method before the OrderBy method. This is because it will first filter results and then it will sort the filtered result which will improve the performance of the application. The following example shows the above using LINQ OrderBy and Where Method with both Query and Method Syntax.



1. How to Create and use Own Comparer with LINQ OrderBy Method in C#?

It is also possible to create and use our own Comparer with LINQ OrderBy Method. For this, we need to use the Second Overloaded version of the OrderBy Method which takes the IComparer parameter. For a better understanding, please have a look at the below example.

