What is Linq OrderByDescending Method in C#?

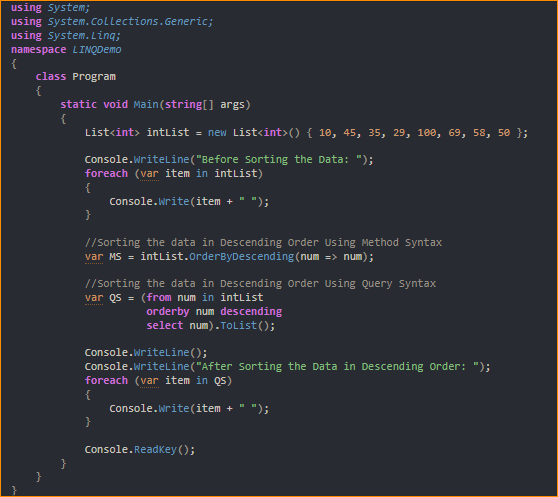
The LINQ OrderByDescending method in C# is used to sort the data in Descending order. The point that you need to remember is, the OrderByDescending method is not going to change the data, it is just changing the order of the data. Like the OrderBy method, you can also use the OrderByDescending method on any data type such as string, character, float, integer, etc. There are two overloaded versions of the OrderByDescending Method available inside LINQ. They are as follows.

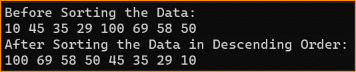


The one and only difference between these two overloaded versions are that the second overloaded version takes the IComparer parameter which basically compares the keys when we are creating our custom comparer. Let us understand the use of the LINQ OrderByDescending method in C# using both query syntax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

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

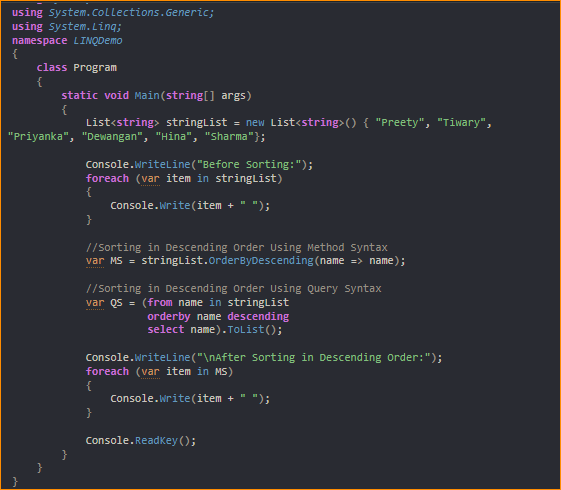
Let us see an example to understand how the LINQ OrderByDescending Method works with Primitive Data Types such as Integer using C#. In the below example, we have a collection of integer data. And we need to sort the data in descending order. For this, we are using the LINQ OrderByDescending method using both Method and Query Syntax.





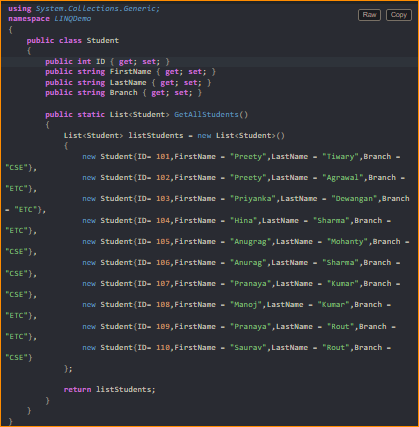
1. LINQ OrderByDescending Method with String Data Type in C#

Let us see an example to understand how the LINQ OrderByDescending Method works with String Data Type using C#. In the below example, we have a collection of string data i.e. a collection of names. We then sort the data in descending order using the LINQ OrderByDescending method with both Method and Query syntax. In the Query Syntax, while we are sorting the data in descending order then the use of descending operator is mandatory. If we omit the descending keyword, then by default it is going to be ascending.

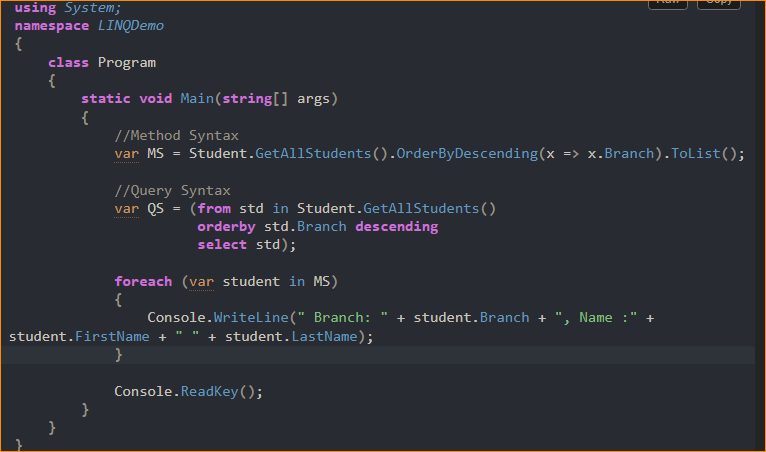


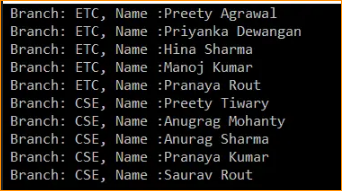
1. LINQ OrderBy Method with Complex Data Type in C#:

Let us see how the LINQ OrderByDescending Method works with Complex Data Types in C# with 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 descending order. To do so, modify the Main method of the Program class as follows. In the below code, we are Sorting the data in descending order based on the student Branch using LINQ OrderByDescending Method with both Method and Query Syntax.





1. Linq OrderByDescending Method with Filtering Method

Now we need to fetch only the ETC branch students and then we need to sort the students based on the FirstName in Descending order. The most important point that you need to remember is, you need to use the Where Extension method before the OrderByDescending 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 OrderByDescending Method and LINQ Where Method with both Query and Method Syntax.

