1. What is the LINQ Skip Method in C#?

The LINQ Skip method **bypasses a specified number of elements in a sequence and then returns the remaining elements.** This method is a part of the System.Linq namespace and can be used with any IEnumerable<T> or IQueryable<T> collection.

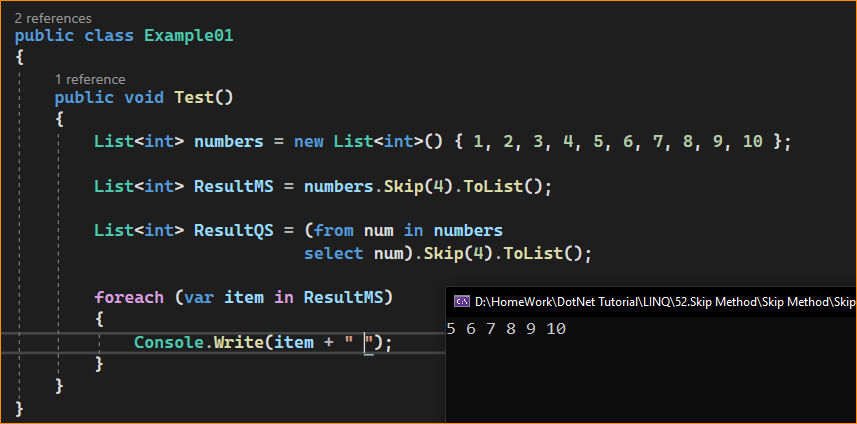
The LINQ Skip Method in C# is used to skip or **bypass the first n number of elements from a data source or sequence and then returns the remaining elements from the data source as output.** Here, n is an integer value passed to the Skip method as a parameter. Only one version is available for the LINQ Skip method.



As you can see in the above signature, the LINQ Skip method takes one integer count parameter and will skip that number of elements from the beginning of the data source. If the data source is null, it will throw an exception. We can use the LINQ Skip method with both Method and Query Syntax. The most important point that you need to remember is that it will not make any changes to the positions of the elements in the original data source.

1. Examples to Understand LINQ Skip Method in C#:

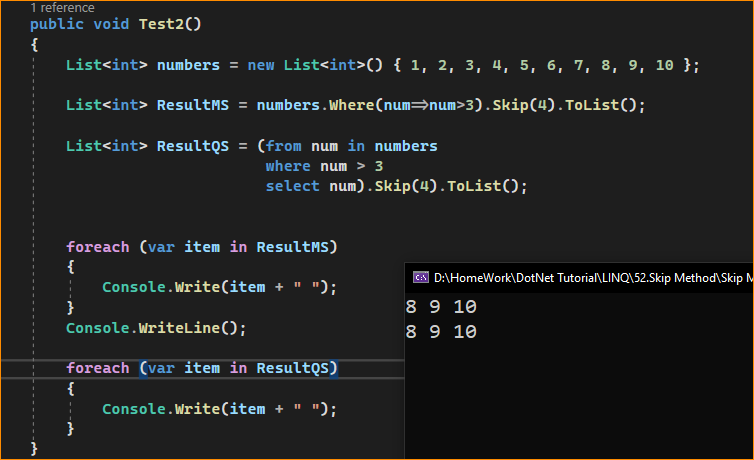
Let us see an example to understand the LINQ Skip Method in C# using both Method and Query Syntax. Please have a look at the following example. In the following example, we created one integer collection which contains 10 elements. Here, we need to skip the first four elements, and then we need to retrieve the remaining elements using the LINQ Skip method.



Note: The Skip method is often used in pagination scenarios where you want to skip several records corresponding to the previous pages and return the next set of records for the current page. It’s also used when you’re interested in discarding a portion of the sequence but processing the rest.

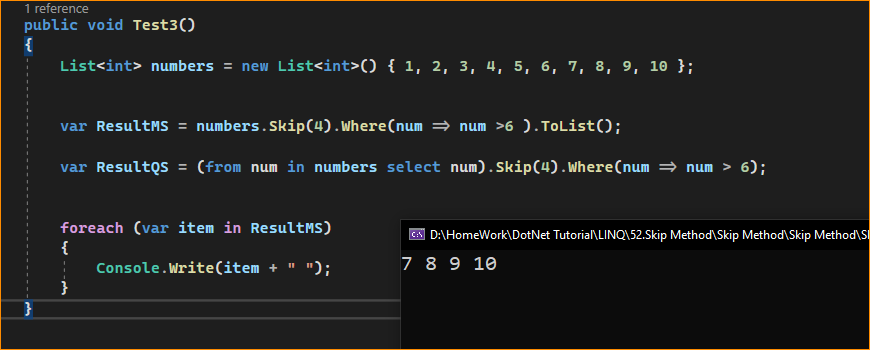
1. LINQ Skip Method with Filtering Method in C#:

Now, let us understand how we can use the LINQ Skip method along with the LINQ Where Extension Method and what is the better approach to use these two methods. Our requirement is to skip the first four elements and return the remaining elements from the collection but the condition is that the elements should be greater than 3. In such cases, first, we need to filter the data using the LINQ Where Extension Method, and then we need to skip the first four elements using the LINQ Skip Method. For a better understanding, please have a look at the following example.



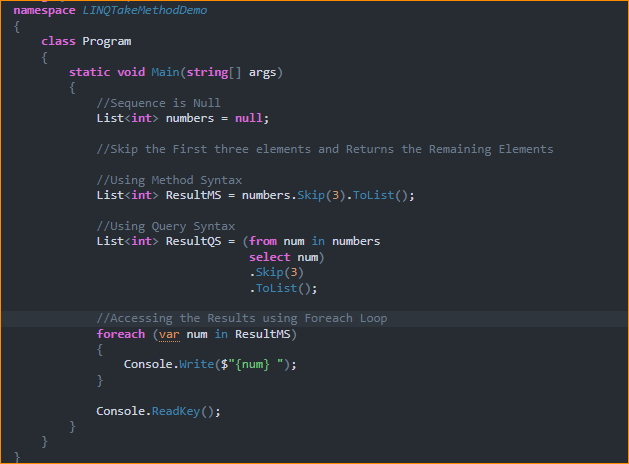
1. Applying Filtering After the Take Method in C#:

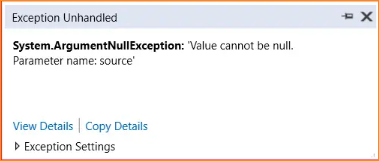
If you apply the Filtering Method, i.e., the Where Method after the Skip method, you will not get the result as expected. For a better understanding, please look at the following example and see the output. Our requirement is to skip the first four elements and return the remaining elements with the condition that the number should be greater than 4. So, our expected output will be 8, 9, and 10.



What happens when the Data Source is null?

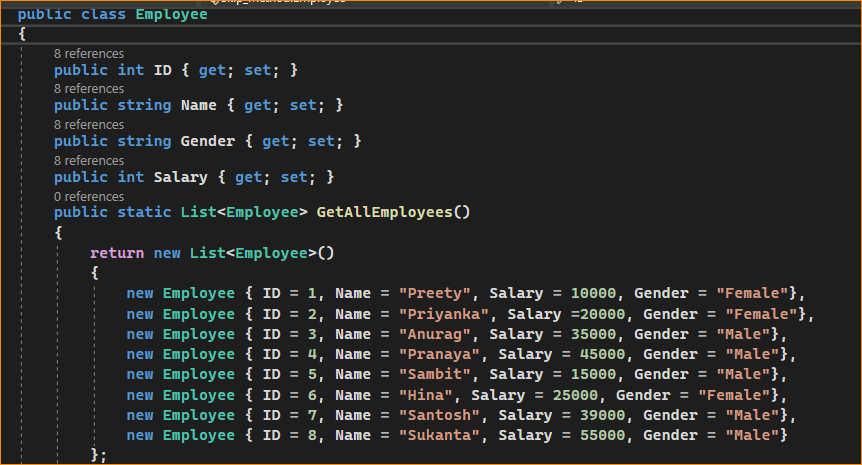
When we apply the Skip Method on a null data source, we will get an exception, i.e., ArgumentNullException, as shown in the example below.





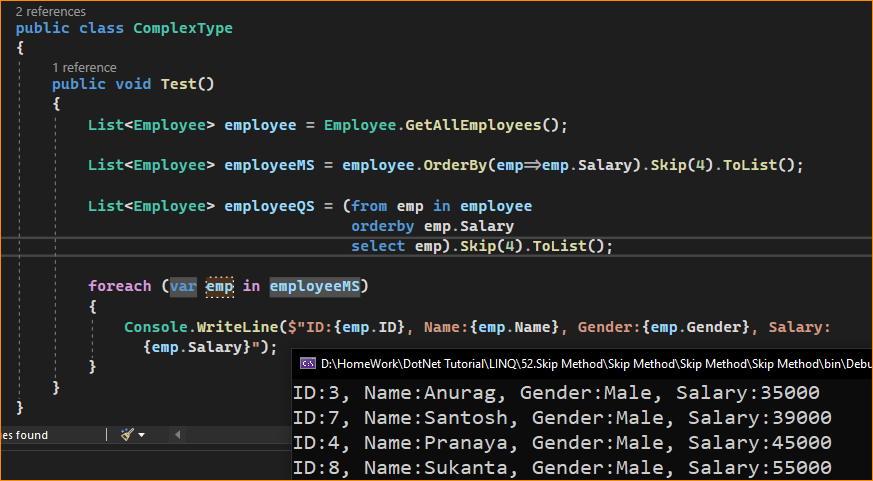
1. Example to Understand LINQ Skip Extension Method with Complex Types in C#:

Let us see how to use the LINQ Skip Extension Method with Complex Data Types using C#. We are going to use the following Employee class. So, create a class file named Employee.cs and copy and paste the following code. As you can see, here we created the Employee class with four properties, i.e., ID, Name, Gender, and Salary. Here, we have also created one method which will return the list of all employees which will be going to our data source.



1. Business Requirement: LINQ Skip Method with OrderBy Method

Our Business Requirement is to skip the first four employees getting the lowest salary. For this, we need to use the OrderBy method on the data source and we need to sort the employees in ascending order by their salary. Then, we need to apply the Skip method and pass the count value as 4 as we need to skip 4 employees. The following example exactly does the same thing.



1. Points to Rember When Working With LINQ Skip Method

Key points to keep in mind when using the Skip method:

* The Skip method is useful when you need to skip a specific number of elements from the beginning of a sequence, especially in scenarios like paging where you want to retrieve a subset of data.
* You can combine Skip with other LINQ methods, such as Take, to implement complex data retrieval and manipulation operations.
* If the count parameter is greater than or equal to the number of elements in the sequence, the Skip method will return an empty sequence.
* The Skip method does not modify the original sequence; it returns a new sequence with the skipped elements.
* The source sequence type and elements in the sequence must match the type specified for the generic parameter of the Skip method.
* The Skip method can be used with any collection or sequence that implements the IEnumerable<T> interface, including arrays, lists, and query results from databases.

1. When to Use LINQ Skip Method in C#?

The LINQ Skip method is useful in various scenarios where you need to bypass a certain number of elements in a collection. Here are some common situations where Skip might be the right choice:

* Pagination: In web and desktop applications, when implementing pagination controls, you would Skip the items from the previous pages and then Take the number of items for the current page.
* Data Processing: When processing data in chunks, you might Skip processed items and continue with the next set of items.
* Performance Optimization: If the initial items in a sequence are not needed or are known to be irrelevant for a particular operation, Skip can be used to start processing from a certain index, potentially reducing execution time.
* Sequential Access: When you need to access elements in a sequence starting at a specific index, Skip allows you to fast-forward to that point.
* Streamlining Data: In a data stream or when reading from a file, you may want to Skip over a header or some predefined number of lines or records.

The LINQ Skip method is valuable when you need to skip elements from the beginning of a sequence or collection to work with a specific subset of data, such as implementing paging, data sampling, or custom data transformations.