1. LINQ Repeat Method in C#:

In C#, the Repeat method is a part of LINQ (Language Integrated Query) and resides in the System.Linq namespace. It is used to generate a sequence that contains one repeated value over a specified number of iterations. The LINQ Repeat Method in C# generates a sequence or a collection with a specified number of elements, each containing the same value. The following is the signature of the LINQ Repeat Method.

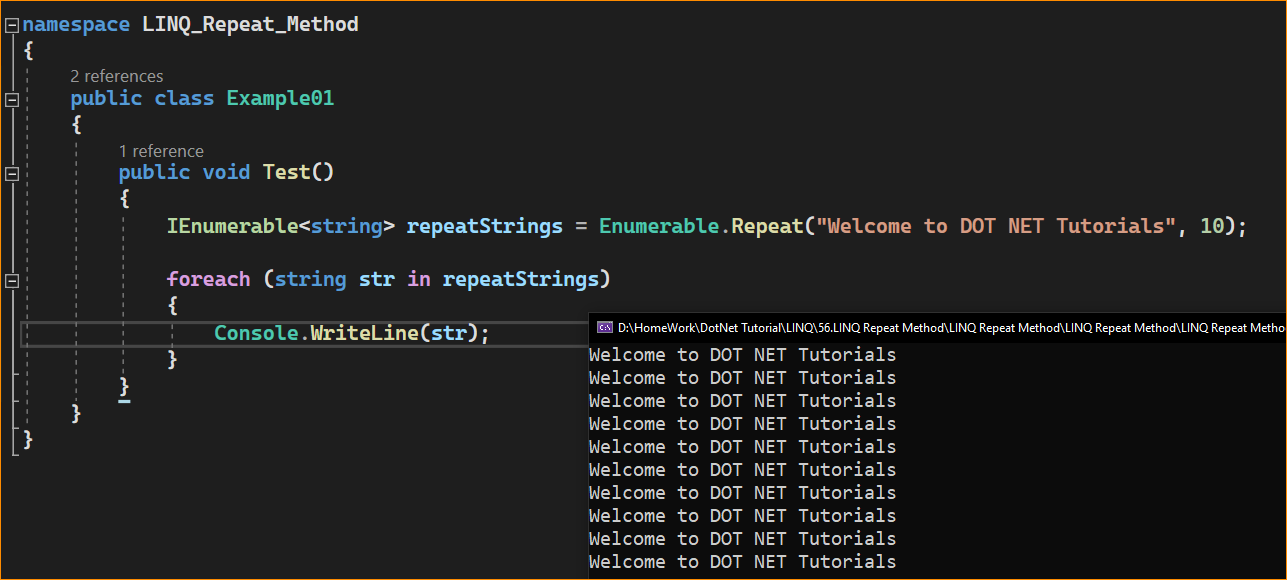


The Repeat method tales 2 integer parameters. The first parameter (i.e., TResult element) specifies the value to be repeated. The second parameter (i.e., int count) specifies the number of times to repeat the value. The return type of this method is IEnumerable<TResult>, which will contain the repeated values. Here, TResult specifies the data type of the value that will be repeated in the result sequence. When the count value is less than 0, it throws ArgumentOutOfRangeException.

* element: The value to be repeated.
* count: The number of times to repeat the value.

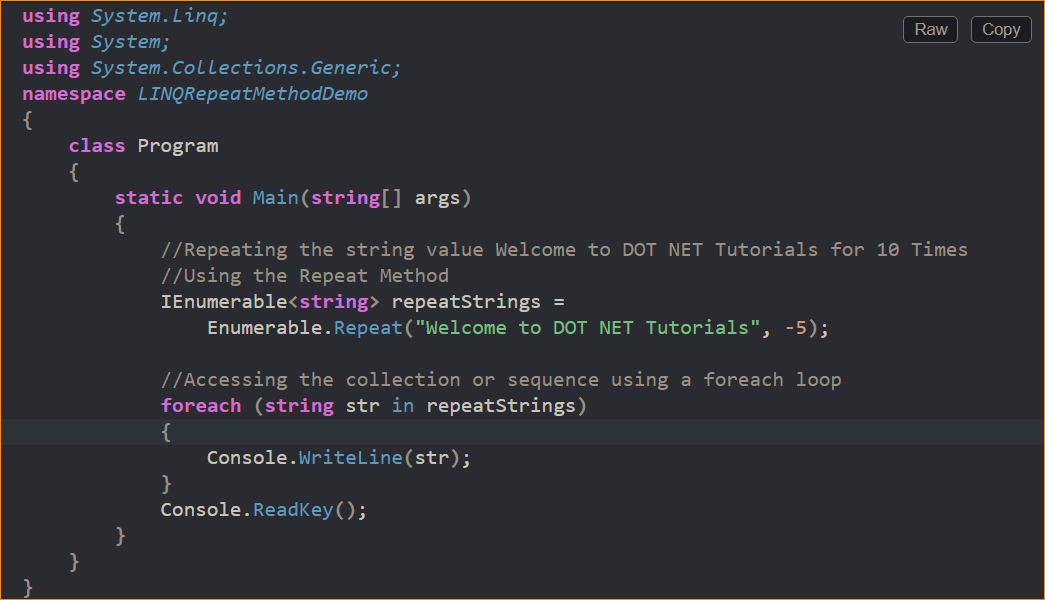
1. Example to Understand LINQ Repeat Method in C#:

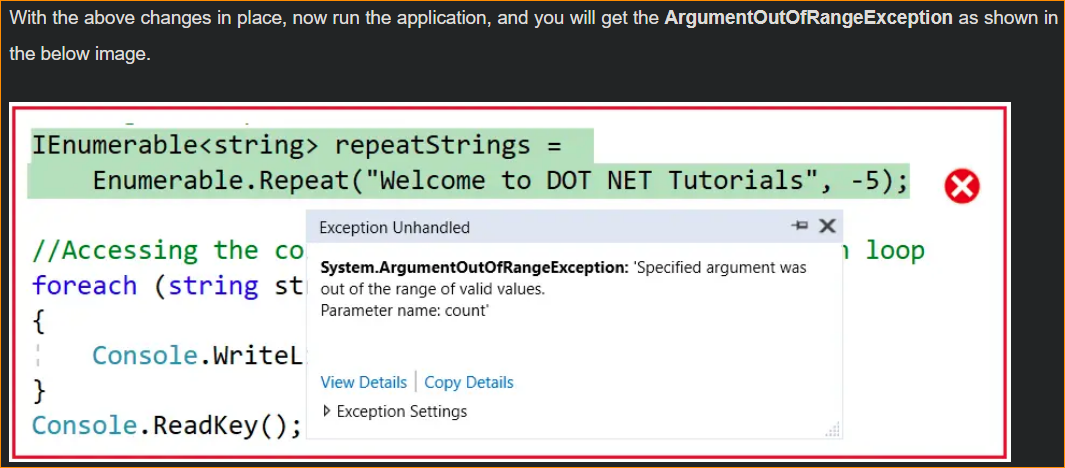
Let us see an example of the LINQ Repeat Method in C#. The following example shows how to use the Repeat method to generate a sequence of repeated values. In the example below, we repeat the string “Welcome to DOT NET Tutorials” 10 times using the Repeat method. As the data type of the value is a string, it will return IEnumerable<string>. Here, TResult is a string. No operator call repeat is available in LINQ to write the Query Syntax. So, it only supports the Method Syntax.



1. What happens if we pass the count as a Negative Number to the Repeat Method in C#?

If we pass the count value as a Negative Number to the Repeat Method, it will throw ArgumentOutOfRangeException. For a better understanding, please have a look at the following example. Here, we are passing the count value as -5.





Note: The Repeat method is implemented using the Deferred Execution. So, the immediate return value is an object which stores all the required information to perform an action. The query represented by this method is not executed until the object is enumerated by calling its GetEnumerator method directly or using a for each loop.

1. When to use the LINQ Repeat Method in C#?

The Enumerable.Repeat method in C# LINQ is handy when generating a sequence of repeated values of the same element. It’s particularly useful when creating collections or sequences with consistent or default values. Here are some common situations where you might use the Repeat method:

**Initializing Collections with Default Values:**

Repeat can initialize arrays, lists, or other collection types with default or initial values. For example, creating an array of zeros or a list of empty strings.

int[] zeros = Enumerable.Repeat(0, 10).ToArray();

Use Cases:

* Filling: When you need to initialize a collection with a default or initial value, Repeat can create a list with the same value for each element, which can then be modified.
* Placeholder Elements: Creating a sequence with placeholder values that will be replaced later in the code.

**Generating Test Data:**

When testing your code, you might need sample data with repeated values. Repeat allows you to create mock data easily.

var mockData = Enumerable.Repeat(new TestData(), 100).ToList();

Use Cases:

* Testing: For unit tests, you must create a collection of items with the same value to test bulk operations or performance.
* Mock Data: Generating mock data for testing or design purposes, such as populating a UI control with dummy data to see how it looks.

