1. What is Filtering?

Filtering is nothing but the process to get only those elements from a data source that satisfied the given condition. It is also possible to fetch the data from a data source with more than one condition as per our business requirements. For example:

* Employees having a salary greater than 50000.
* Students Having Marks greater than 80% from a particular batch.
* Employees having experience of more than 6 Years and the department is IT, etc.

1. What are the Filtering Methods Available in LINQ?

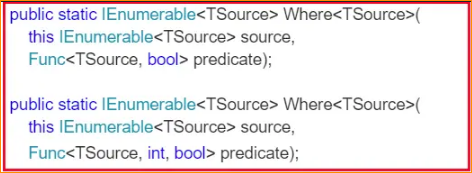
There are two methods provided by LINQ in C# which are used for filtering. They are as follows:

* **Where Method**
* **OfType Method**

1. LINQ Where Filtering Operator or Method:

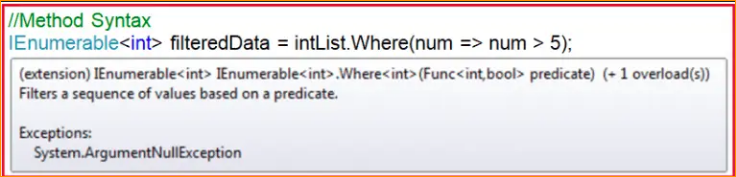
The standard query operator “Where” comes under the Filtering Operators category in LINQ. We need to use the Where standard query operator or method in LINQ when we need to filter the data returned from a data source based on some condition(s) just like we did in SQL using the Where clause. So in simple words, we can say that it is used to filter the data from a data source based on some condition(s).

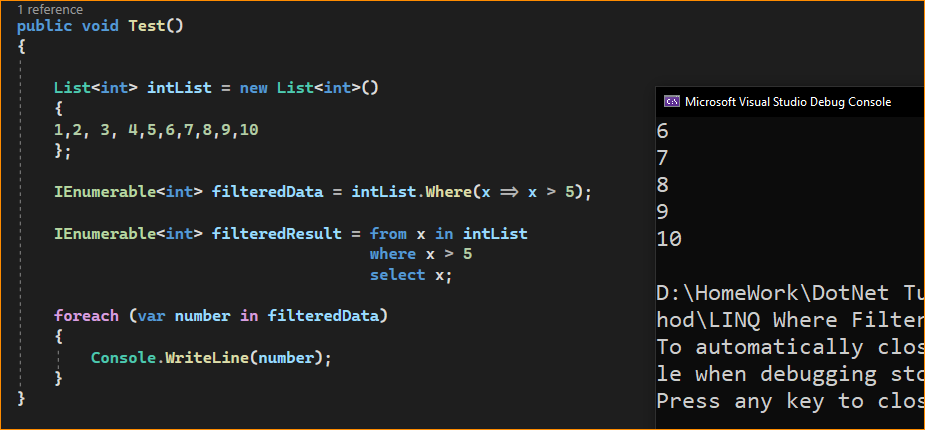
The “Where” Method always expects at least one condition and we can specify the condition(s) using predicates. The conditions can be written using the ==, >=, <=, &&, ||, >, <, etc. symbols. There are two overloaded versions of the “Where” Method available in LINQ. They are as follows:



As you can see in the above signatures, the methods are implemented as extension methods on IEnumerable<T> interface. The methods accept a predicate as a parameter. So let us first understand what a predicate is.

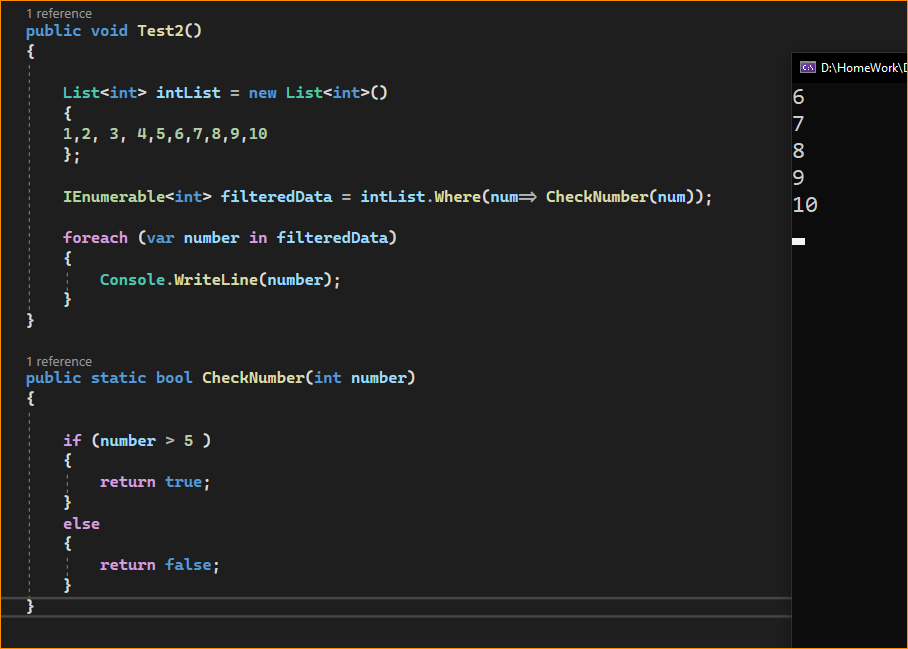
A Predicate is nothing but a function or technically you can say a delegate that is used to test each and every element for a given condition. Let us understand this with an example. In the below example, the Lambda expression (num => num > 5) runs for each and every element present in the “intList” collection. Then it will check, whether the number is greater than 5 or not. If the number value is greater than 5, then a boolean value true is returned otherwise false. In the below example, I am showing both Method and Query Syntax.





Same happen in here –

Here num – 1,2,3 ….9



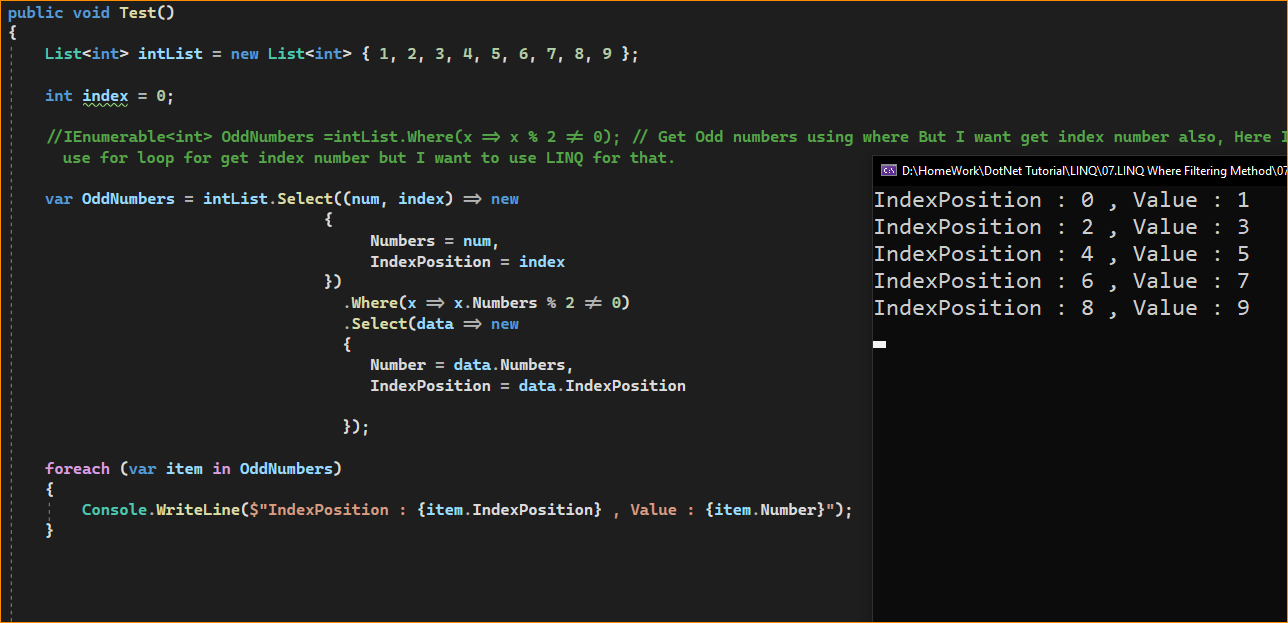
1. Example to Understand the Second Overloaded Version of the Where Method in LINQ:

In the second overloaded version of the “Where” extension method, the int parameter of the predicate function represents the index position of the source element.



Let us see an example to understand this. Here we need to filter only the odd numbers i.e. the numbers which are not divisible by 2. Along with the numbers we also need to fetch the index position of the number. The index is 0 based.

1. Let us see an example to understand this. Here we need to filter only the odd numbers i.e. the numbers which are not divisible by 2. Along with the numbers we also need to fetch the index position of the number. The index is 0 based.
2. Using Method in LINQ s

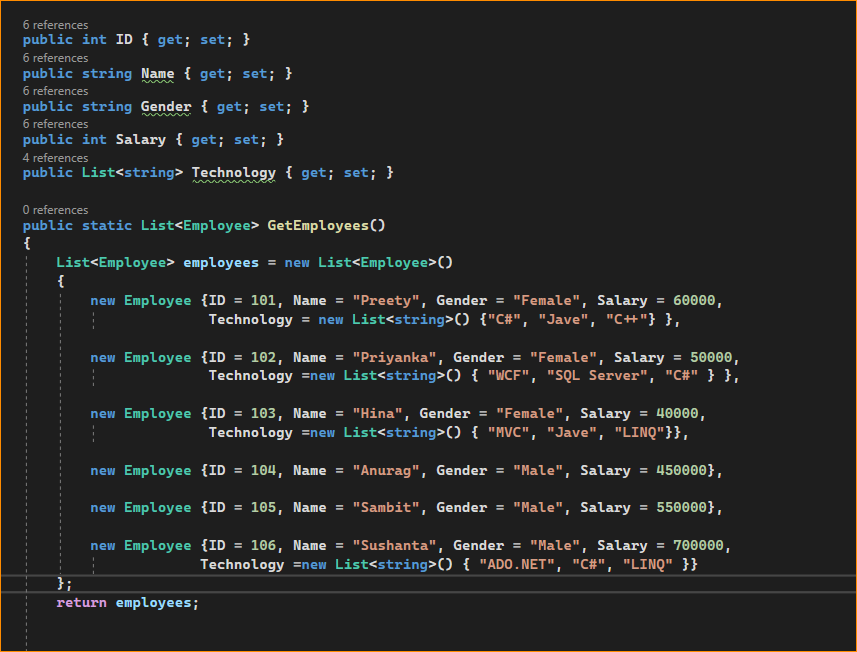


1. LINQ query syntax

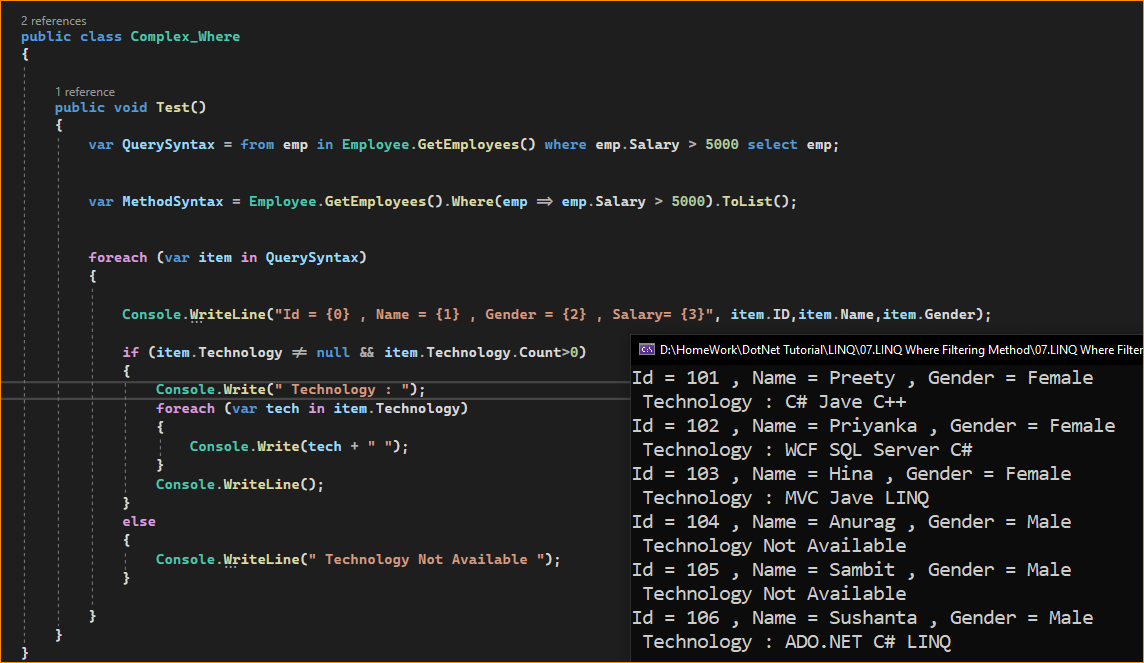


1. Complex Example to Understand LINQ Where Extension Method using C#:

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

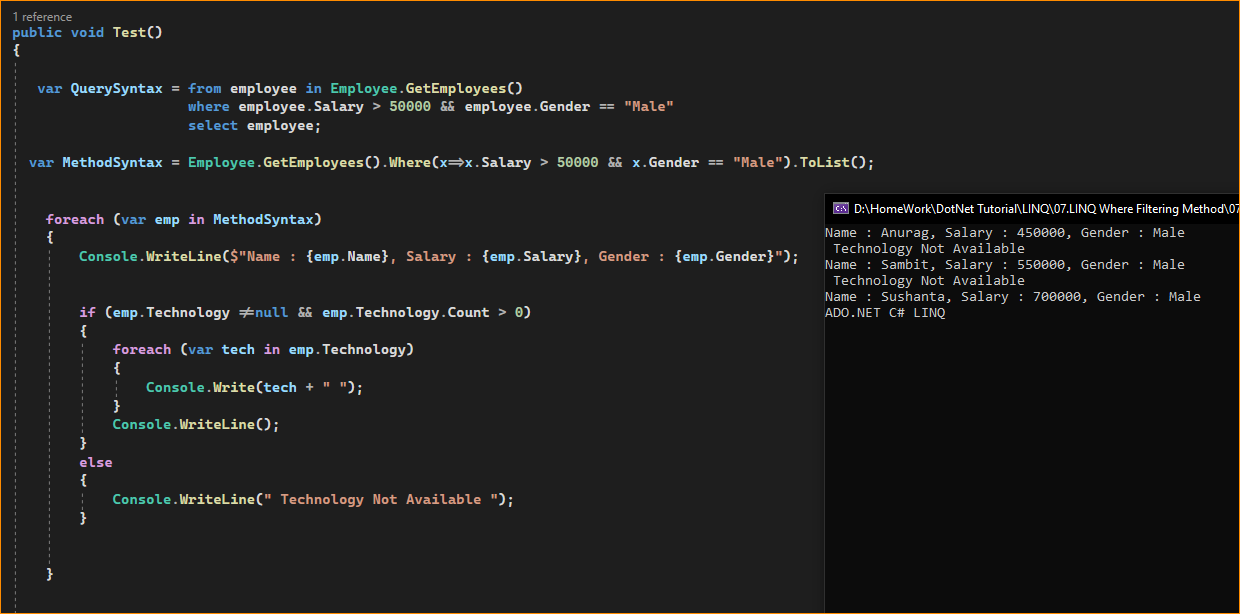


we need to fetch all the employees whose salary is greater than 50000. For this, we need to use the LINQ Where Extension Method and we need to specify the condition as emp => emp.Salary > 50000 using Method Syntax and employee.Salary > 50000 using Query Syntax which is shown in the below example.



1. Example to Understand How to Specify Multiple Conditions using LINQ Where Method in C#

We need to fetch all the employees whose gender is Male and also whose Salary is greater than 500000. So, here we have two conditions. The first condition is Gender = Male and the second condition is Salary > 500000. If we have more than one condition, then we need to use && (AND) or || (OR) logical Operators based on our requirement. Here, we are going to use the AND (&&). That means if both the conditions satisfy then only return the data.



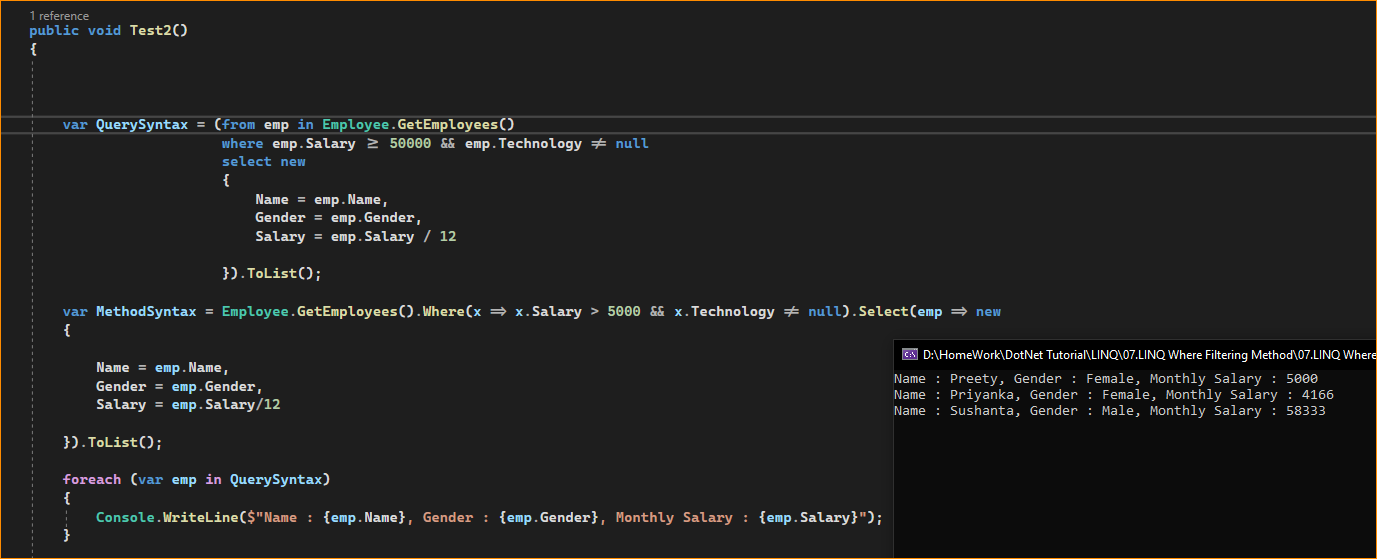
1. Complex Example to Understand LINQ Where Method:

Let us see a more complex example to understand the LINQ Where Extension Method. Now, we are going to provide Multiple conditions with the custom operations and project the data to an anonymous type. The following is our business requirement.

We need to fetch all the employees whose salary is greater than or equal to 50000 and technology should not be null. And we need to project the following information to an anonymous type.

1. Name as it is
2. Gender as it is
3. MonthlySalary = Salary / 12

The Complete Example Code is Given Below. In the below example, I am showing both the Method and Query syntax with the Where Extension Method to achieve the above requirement.



9. Example to Fetch Elements along with their Index position using Where Method in C#

Now, we need to fetch all the employees whose Gender is Male and whose Salary is greater than 500000 along with their index position to an anonymous type.

