

By OnlineInterviewQuestions.com

Latest LINQ Interview Questions and Answers

What is LinQ?

LINQ is the acronym for Language Integrated Query. It is a collection of standard query operators that adds native data querying facilities to .NET framework Languages like C#, VB.NET. LINQ is used as a medium to fulfill the gap between the world of objects and the world of data. In other words, LINQ is a programming model that is composed of general-purpose standard query operators. This allows you to work with data irrespective of the data source.

LINQ was designed by **Anders Hejlsberg** and was introduced in Visual Studio in 2008. LINQ queries can be written for numerous data types, and you can write queries without having the knowledge of query languages like SQL, XML, etc. There has been a huge scope in the field of LINQ, and more and more people are applying for LINQ related jobs. If you're too planning to sit for a **LINQ interview**, then jump straight into the questions given below and crack the LINQ interview nuts like an expert.

Q1. What do you understand by LINQ?

Language Integrated Query (LINQ) is the collection of standard query operators that adds query facilities into .NET framework languages like C#, VB.NET, etc. LINQ also represents the set of method names with the translation rules that can be used by the compiler for converting fluent-style query expressions into expressions that uses these method names, anonymous types and lambda expressions. LINQ can be used to extract data from arrays, XML documents, relational databases and other third-party data sources.

The LINQ is of the following types:

- LINQ to objects refers to the use of LINQ queries without the use of an intermediate LINQ provider such as LINQ to SQL.
- LINQ to XML which was formerly known as XLINQ
- LINQ to SQL which was formerly known as DLINQ
- LINQ to Datasets
- LINQ to Entities

Q2. What are the three main components of LINQ?

The three main components of LINQ are:

- **Standard Query Operators:** These are the extension methods that form the LINQ patterns. It operates

on sequences and forms an API that enables querying of any .NET array. The Standard Query Operators in LINQ allows you to perform functions like determining if a value exists in the sequence, summation over a sequence, etc. The query operators supported by the API are:

- **Language extensions:** LINQ defines optional language extensions when LINQ is primarily implemented as a library for .NET framework. This makes queries a first-class language construct and also provides syntax for writing queries.
- **LINQ Providers:** These are set of classes that generate a method that executes the same query against a particular data source. The different providers define the different flavors of LINQ:
 - LINQ to SQL
 - LINQ to XML
 - LINQ to Objects

Q3. Explain what is select clause and where clause in LINQ?

Select clause and Where clause in LINQ

- **Select clause:** the select clause selects the interesting aspects of the element by performing a projection on the collection. First, the user supplies an arbitrary function in the name of lambda expression which projects the data members. Then, this function is passed as a delegate.
- **Where clause:** The Where clause is used to add some additional filters to the query. It allows the definition of a set of predicate rules that are evaluated for each object. The objects that do not match the rules are altered away. Then, the predicate is supplied as a delegate to the operator.

Q4. How is LINQ useful than Stored Procedures?

LINQ is useful than stored procedures in the following ways:

- **Deployment:** In LINQ, deployment becomes easier as everything is compiled into a single DLL. However, in Stored Procedures, an additional script has to be provided.
- **Debugging:** since LINQ is a part of .NET languages, queries can be debugged using the visual studio's debugger. In the case of Stored Procedures, there is difficulty in debugging.
- **Type safety** at the compile time, all the queries errors are checked as LINQ is type safe. However, stored procedures are not type-safe, so there is a difficulty in checking the errors.

Q5. What do you understand by DataContext class? How is it related to LINQ?

After adding **LINQ to SQL classes** and after opening the O/R Designer, there is an empty design surface that represents the empty DataContext class that is ready to be configured. This class is a LINQ to SQL class that acts as a channel between a SQL Server database and the LINQ to SQL entity class. This class contains information regarding the methods for connecting to a database and also manipulating the data in the database. DataContext classes are configured with the connection data that is provided by the first item.

Q6. What is PLINQ?

PLINQ is the acronym for **Parallel Language Integrated Query**. It is used to execute a query by using multiple processors, and it is a parallel implementation of LINQ. At the places where data grows rapidly and where data is heterogeneous, PLINQ ensures scalability of software on the parallel processors. PLINQ is more preferred over LINQ because it supports all the operators of LINQ. It also makes effective use of all the processors in the system, and it runs several LINQ queries simultaneously. **PLINQ** also ensures that the queries run quickly as it uses parallel execution. Due to its parallel performance, PLINQ enjoys major performance improvement, unlike some legacy codes that require too much time to execute.

Q7. Differentiate between Select() and SelectMany() method in LINQ.

The **Select()** and the **SelectMany()** are both used to produce result values from the sources of data. However, the difference is the result set. On the one hand, the select clause produces one result value for every source value, and the result value is the collection that contains the same number of elements from the query. On the other hand, the **SelectMany()** produces a single result that keeps a concatenated collection from the query.

Q8. What is the basic syntax of a LINQ query in Visual Basic as well as in C#?

In **Visual Basic**, From clause is used to start the basic syntax of the LINQ query. However, it ends with Select or Group By clause. If you want to perform additional functions like filtering data or generating data in a specific order, you can use another clause like Where, Order By, Order By Descending, etc.

In **C#**, the basic syntax starts with the From clause and terminates with Select or Group By clause. If you want to perform various functions like filtering data or generating data in a specific order, you can use another clause like Where, Order By, Order By Descending, etc.

Q9. How are standard Query Operators useful in LINQ?

Standard Query Operators are useful in **LINQ** in following ways-

- It helps to get the total count of elements in the collection.
- It helps in ordering the results of the collection.
- It helps in grouping and computing average.
- It joins two collections based on the matching keys.
- It filters the results out.

Q10. Enlist few advantages of LINQ?

LINQ offers several advantages, and one of its main advantages is that it enables the developers to express declaratively. Some other advantages are:

- The development time gets reduced in LINQ as the writing code is quite faster.
- In LINQ, it is easy to find errors during design time as it offers syntax highlighting
- LINQ encourages joining of several data sources in a single query.
- It also offers the facility to break complex problems into a set of short queries which are easy to debug.
- It is possible to use LINQ knowledge for querying new data sources types.

Q11. Categorize LINQ Standard Query Operators based on functionality.

LINQ operators can be categorized based on functionality in following ways-

- **Filtering operators**- it is used to restricting the result set in a way that it has only selected elements that satisfy a condition.
- **Join operators**- it is an operation where data sources with difficult to follow relationship with each other are targeted.
- **Sorting operators**- it sorts the elements of a sequence on some attributes.
- **Conversions**- it is an operator that changes the type of input objects, and they are also used in various applications.
- **Aggregations**- it helps to perform any type of aggregation and also creates custom aggregations.

Q12. What do you understand by LINQ to XML?

LINQ to XML has made loading of XML documents, querying and document modification easier. While using LINQ to XML, it is possible to save XML documents existing in memory to disk. It also eliminates the need for a developer to learn XML query language. There are 19 necessary classes to work with XML. To name a few- XAttribute, XCDATA, XComment, XElement, XName, XObject, object change. XTest, XDocument, XDocumentType, etc.

Q13. What are the important points to remember while using lambda expressions?

The important points to remember while using lambda expressions are:

- A lambda expression can return a value. It may have parameters.
- With the help of a feature called closure, it is possible to access variables that are present outside of lambda expression.
- We should use closer feature very cautiously to avoid any kind of problems.
- Lambda expressions are not used on the operator's left side.
- Curly brackets are not required if there is a single statement. However, if there are multiple statements, curly brackets and return values are essential to writing.

Q14. Differentiate between statement lambda and expression lambda.

Expression lambdas are generally used for constructing expression trees whereas **Statement lambdas** cannot be used to create expression trees.

Q15. What are benefits of using LINQ in Dataset?

The main motive to use LINQ in a dataset is to retrieve strongly typed queries in a dataset. LINQ is used when we want to combine values from two different datasets, or we want a unique value from the dataset.

You can use SQL queries to populate the dataset, but you cannot use SQL queries to retrieve values. If you want to retrieve particular values, you have to use ADO.NET functionalities. In the case of LINQ, it gives a clearer way of querying the dataset. LINQ also provides additional features as compared with ADO.NET.

Q16. Differentiate between Conversion Operator “ToDictionary” and “IEnumerable” of LINQ.

"**IEnumerable**" and "**ToDictionary**" conversion operators are used to solve the conversion type problems. "ToDictionary" conversion operator is a part of Dictionary (k, T).

The "keySelector" is used to recognize the key of each item, and "elementSelector" is used to extract each item. "IEnumerable" simply returns the source sequence as an object of IEnumerable <T>.

Q17. Write the basic steps to execute LINQ query?

The LINQ query can be executed in following way:

- First, you have to obtain the data source. This can be an XML file or an SQL database.
- Then, you need to create a query
- Once you've created the query, it's time to execute the query.

Q18. What is a LinqDataSource control?

If you want to use **LINQ in an ASP.NET** webpage, then LinqDataSource control is a thing for you. It sets the properties in the markup text, and you can use the control to retrieve and modify data. LinqDataSource control can be used to declaratively bind other ASP.NET controls on a page to a data source, and as such it is similar to SQL Datasource and ObjectDataSource controls. However, there lies a difference in the sense that instead of binding directly to a database, LinqDataSource control binds the LINQ enabled data model.

Q19. What do you understand by Expression Tree?

Lambda expressions are widely used in the Expression tree which gives away code in a data structure which resembles a tree. Every node in the tree is itself an expression like a method call or binary operation. Statement lambdas which are two or three statements don't help in the construction of expression trees.

Q20. What are the benefits of tier architecture style?

The main benefits of tier architecture style are:

- **Maintenance**- any alteration or modification can be done without affecting the application as a whole because each tier is independent of other tiers.
- **Flexibility**- there is a huge scope of flexibility as each tier is managed and maintained independently.
- **Scalability**- tiers are based on the deployment of layers. So, scalability is easier.

Please Visit OnlineInterviewquestions.com to download more pdfs