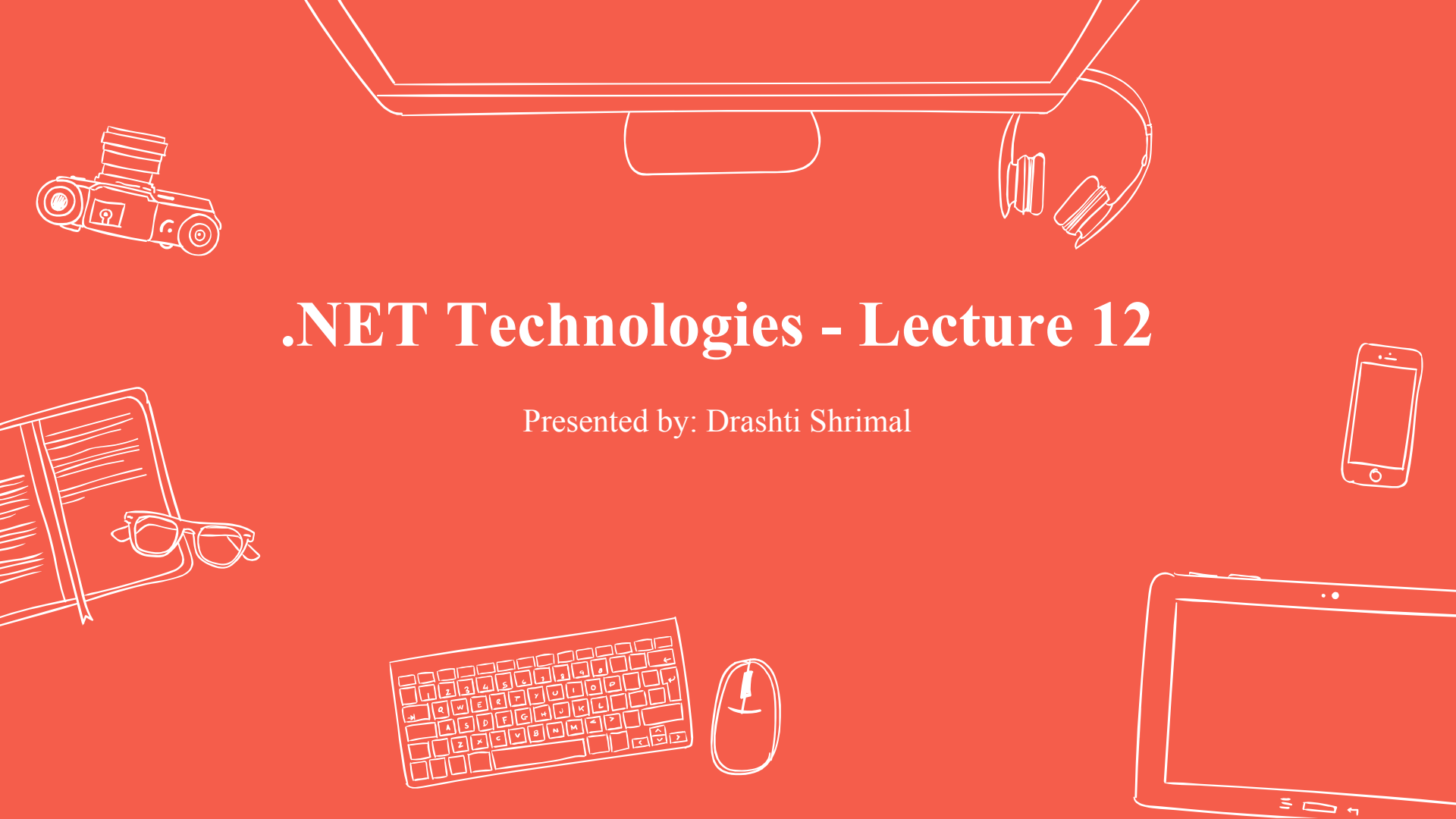

.NET Technologies
MacBook Air



.NET Technologies - Lecture 12

Presented by: Drashti Shrimal

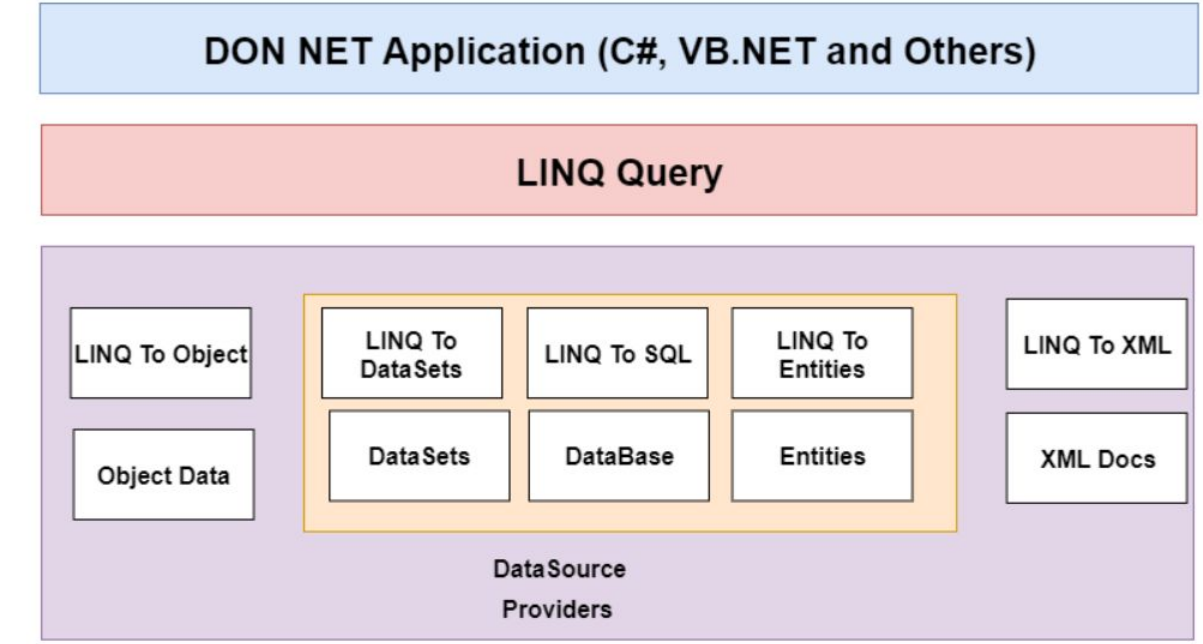
UNIT 2: Topics in this presentation

Topics:

- LINQ Intro
- LINQ Architecture
- LINQ Need
- LINQ advantages
- LINQ disadvantages
- LINQ Query
- LINQ Example

LINQ Intro

- The full form of LINQ is 'Language Integrated Query,' and introduced in .NET Framework 3.5 to query the data from different sources of data such as collections, generics, XML documents, ADO.NET Datasets, SQL, Web Services, etc. in C# and VB.NET.
- LINQ provides the rich, standardized query syntax in a .NET programming language such as C# and VB.NET, which allows the developers to interact with any data sources.
- In C# or VB.NET, LINQ functionality can be achieved by importing the System.Linq namespace in our application.
- Generally, the LINQ contains a set of extension methods which allows us to query the source of data object directly in our code



LINQ Architecture

LINQ Need

- LINQ is simpler, ordered, and higher-level than SQL. When we want to use Querying Database, in most cases, LINQ is a more productive query language than SQL.
- Also, we have the benefits of IntelliSense as the LINQ query is written in behind the code. LINQ has full type checking at compile time so that we can catch any error in compile time itself. In C# or VB.Net to write the query in LINQ is more fun.
- Here, we are taking an example of developing a .NET application, and that application requires data from different data sources.

LINQ Need

- Suppose the application needs the data from SQL Server Database. So, as a developer, to access the data from SQL Server Database, we need to understand ADO.NET and SQL Server-specific syntaxes. If the database is Oracle, then there is a need to understand the SQL Syntax, which is specific to Oracle Database.
- The application also needs the data from an XML document. So, as a developer to work with an XML document, we need to understand the XSLT and XPath queries.
- In the application, there is also a need to manipulate the data (objects) in memory such as List, etc. So, as a developer, there is a need to understand how to work with in-memory objects.

LINQ Need

- LINQ provides the Uniform Programming Model (i.e., familiar query syntax).
- It allows us to work with different data sources by using a standard, or in a unified coding style.
- Hence, we don't require to learn the different syntaxes to query for various data sources.

Advantages of LINQ

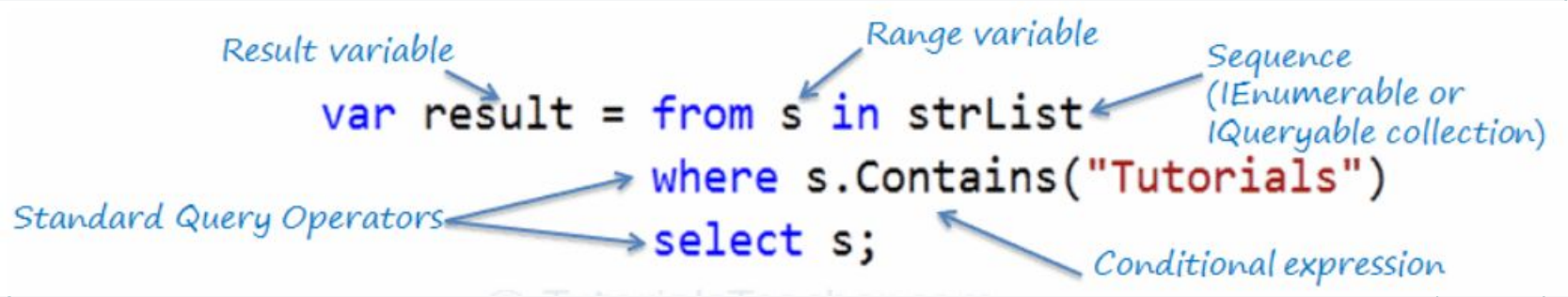
In our applications, the benefits of LINQ are:

- We do not need to learn new query language syntaxes for different sources of data because it provides the standard query syntax for the various data sources.
- In LINQ, we have to write the Less code in comparison to the traditional approach. With the use of LINQ, we can minimize the code.
- LINQ provides the compile-time error checking as well as intelligence support in Visual Studio. This powerful feature helps us to avoid run-time errors.
- LINQ provides a lot of built-in methods that we can be used to perform the different operations such as filtering, ordering, grouping, etc. which makes our work easy

Disadvantages of LINQ

- With the use of LINQ, it's very difficult to write a complex query like SQL.
- It was written in the code, and we cannot make use of the Cache Execution plan, which is the SQL feature as we do in the stored procedure.
- If the query is not written correctly, then the performance will be degraded.
- If we make some changes to our queries, then we need to recompile the application and need to redeploy the dll to the server.

LINQ Query Syntax



Example

```
        // string collection
        IList<string> stringList =
new List<string>() {
        "C# Tutorials",
        "VB.NET Tutorials",
        "Learn C++",
        "MVC Tutorials" ,
        "Java"
    };
```

```
// LINQ Query Syntax
        var result = from s in
stringList
   where s.Contains("Tutorials")
   select s;
  foreach (var u in result)
   {
       Console.WriteLine(u);
   }
```