Q: What is LINQ?
ANS: It stands for Language Integrated Query. LINQ is collection of standard query operators that provides the query facilities into .NET framework language like C# , VB.NET.

How LINQ is beneficial than Stored Procedures?
ANS: There are couple of advantage of LINQ over stored procedures.
1. Debugging - It is really very hard to debug the Stored procedure but as LINQ is part of .NET, you can use visual studio's debugger to debug the queries.
2. Deployment - With stored procedures, we need to provide an additional script for stored procedures but with LINQ everything gets complied into single DLL hence deployment becomes easy.
3. Type Safety - LINQ is type safe, so queries errors are type checked at compile time. It is really good to encounter an error when compiling rather than runtime exception!

Q: Why Select clause comes after from clause in LINQ?
ANS: The reason is, LINQ is used with C# or other programming languages, which requires all the variables to be declared first. From clause of LINQ query just defines the range or conditions to select records. So that’s why from clause must appear before Select in LINQ.

Q: What is the extension of the file, when LINQ to SQL is used?
ANS: The extension of the file is .dbml

Q: What is the LINQ file extension that interacts with Code Behind's objects.
ANS: its .dbml

Q: Why can't datareader by returned from a Web Service's Method
ANS: Cos, it's not serializable

Q: What is the use of System.XML.XLinq.dll?
ANS: System.XML.XLinq.dll contains classes to provide functionality to use LINQ with XML.

Q: What is the use of System.Data.DLinq.dll?
ANS: System.Data.DLinq.dll provides functionality to work with LINQ to SQL.

Q: Which assembly represents the core LINQ API?
ANS: System.Query.dll assembly represents the core LINQ API.

Q: What is the benefit of using LINQ on Dataset?
ANS: The main aim of using LINQ to Dataset is to run strongly typed queries on Dataset.

Suppose we want to combine the results from two Datasets, or we want to take a distinct value from the Dataset, then it is advisable to use LINQ.

Normally you can use the SQL queries to run on the database to populate the Dataset, but you are not able to use SQL query on a Dataset to retrieve a particular values. To get this you need to use ADO.NET functionalities. But, in case of LINQ, it provides more dignified way of querying the Dataset and provides some new features as compared to ADO.NET.

Q: What are the advantages of LINQ over Stored Procedures?
ANS: Below is the three advantages of LINQ over stored procedures.

Debugging - As debug point concern, as LINQ is part of .NET, we can use the visual studio's debugger to debug the queries but it is tough to debug the Stored procedure as it will not support the visual studio debugger.

Deployment - In case of deployment, we need to provide an additional script for stored procedures to execute but in case of LINQ, it will complie into single DLL hence deployment becomes easier.

Type Safety - As LINQ is type safe, the queries errors are type checked at compile time. Better suggest to use LINQ because it helps to encounter an error at the compile time rather than at runtime exception.

Q: What is the disadvantage of LINQ over stored procedures?
ANS: The disadvantage with LINQ is, it is not a precompiled statement where as stored procedures are precompiled. In case of LINQ the queries need to be compile before the execution. So according to this, I can say stored procedures are faster in performance as compared to LINQ.

Q: What are Quantifiers?
ANS: They are LINQ Extension methods which return a Boolean value
1)All
2)Any
3)Contains
4)SequenceEqual

example:
```
int[] arr={10,20,30};
var b=arr.All(a=>a>20);
-------------------------------------------
```
Output:
b will return False since all elements are not > 20.

Q: Difference between XElement and XDocument
ANS: Both are the classes defined by System.Xml.Linq namespace

XElement class
represents an XML fragment
XDocument class represents an entire XML document with all associated meta-data.

example:

```
XDocument d = new XDocument(
new XComment("hello"),
new XElement("book",
new XElement("bookname", "ASP.NET"),
new XElement("authorname", "techmedia"),

)
);
```

Q: When generating database mappings using LINQ to SQL, which tool allows you to create entity classes using a convenient graphical interface?
ANS: This is objective type question, Please click question title for correct answer.

Q: Briefly can you explain the purpose of LINQ providers ?
ANS: They are a set of classes that takes a LINQ query and dynamically generates on the fly query which is executed against a specific data source(sql database, oracle, xml file, array...etc

Q: What is the difference between N-layer and N-tier architecture?
ANS: N-layers of application may reside on the same physical computor(same tier) and the components in each layer communicates with the components of other layer by well defined interfaces.Layered architecture focuses on the grouping of related functionality within an application into distinct layers that are stacked vertically on top of each other.Communication between layers is explicit and loosely coupled.With strict layering, components in one layer can interact only with componentsin the same layer or with components from the layer directly below it.

The main benefits of the layered architectural style are:
Abstraction, Isolation, Manageability, Performance, Reusability, Testability.

N-tiers architectue usually have atleast three separate logical parts,each located on separate physical server.Each tier is responsible with specific functionality.Each tier is completely independent from all other tier, except for those immediately above and below it.Communication between tiers is typically asynchronous in order to support better scalability.

The main benifit of tier achitecture styles are
1.Maintainability. Because each tier is independent of the other tiers, updates or changes can be carried out without affecting the application as a whole.
2.Scalability. Because tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.
3.Flexibility. Because each tier can be managed or scaled independently, flexibility is increased.
4.Availability. Applications can exploit the modular architecture of enabling systems using easily scalable components, which increases availability.

Q: Tell me the exact difference between IQueryable and IEnumerable interface ?
ANS: IEnumerable<T> is applicable for in-memory data querying, and in contrast IQueryable<T> allows remote execution, like web service or database querying.
Q: What is the difference between Count() and LongCount() extension methods in LINQ ?
ANS:

```
public static long display()

    {

        var tempval = (from h in
objDB.tbl_mvc_login

                       select h).Count ();



        return tempval;

    }

public static long display()

    {

        var tempval = (from h in
objDB.tbl_mvc_login

                       select h).LongCount ();



        return tempval;

    }
```

Look carefully to the above methods declared. They both does the same thing but LongCount() has a greater range than Count(). According to MSDN, it has the range from

long.MinValue = -9223372036854775808
long.MaxValue =  9223372036854775807

Q: Which is quite big. Its DotNet Framework type is System.Int64. While count() DotNet Framework type is System.Int32 which has a range from
ANS:
long.MinValue = -2,147,483,648

long.MaxValue =  2,147,483,647

So, next time if you want to count something which is quite big then use LongCount() extension method otherwise use Count().

Q: Can you explain in brief about Aggregate() extension method in LINQ ?
ANS:

```
public static int display()

    {

        int[] numbersArray = { 1, 2, 3, 4, 5 };
```

```
            return numbersArray.Aggregate((x1,
x2) => x1 * x2);

        }
```

output : 120

In the above code, "numbersArray" is an integer array. Here, the first one being the first number or the previous result, and the second one is the second or next number to participate the calculation.

The calculation goes like :-

1 * 1 = 1 (stored in x1)

x1 * 2 = 2 (stored in x1)

x1 * 3 = 6 (stored in x1)

x1 * 4 = 24 (stored in x1)

x1 * 5 = 120 (stored and returned back)

Q: What is the difference between FirstOrDefault() and SingleOrDefault() extension method in LINQ ?
ANS:

FirstOrDefault() = gets the first item that matches a given criteria.

SingleOrDefault() = if you specify this extension method that means you are specifically saying that there can be only one value that matches the criteria. If there are more then 1 value that matches the criteria, throw an exception.

Q: What is the difference between First() and Single() extension methods in LINQ ?
ANS:
•First() - There is at least one result, an exception is thrown if no result is returned.

•Single() - There is exactly 1 result, no more, no less, an exception is thrown if no result is returned.

Q: How to get the Value of attribute from XML using XDocument class?
ANS:
Let us look at the below situation

```
<?xml version='1.0' encoding='UTF-8'?>

  <Countries>

        <State Name = 'Karnataka'>

             <City Name='Bangalore' />

             <City Name= 'Guledgudda' />

             <City Name= 'Hubli' />

             <City Name= 'Tumkur' />

        </State>

   </Countries>
```

The challenge is to get the City Names using XDocument class. The below code will help us to do so

```
string inputXml = @"<?xml version='1.0'
encoding='UTF-8'?>

                    <Countries>

                        <State Name =
'Karnataka'>

                            <City Name='Bangalore'
/>

                            <City Name= 'Guledgudda' />

                            <City Name= 'Hubli' />
                    <City Name= 'Tumkur' />

                      </State>

                    </Countries>";

XDocument countrySource =
XDocument.Parse(inputXml);

//The query

countrySource.Descendants("State").SelectMany
(i =>
i.Elements()).ToList().ForEach(i=>Console.Writ
eLine((string)i.Attribute("Name")));

//Output
Bangalore
Guledgudda
Hubli
Tumkur
```

Q: Explain with an example how to perform group by in LINQ/LAMBDA?
ANS:
Consider the below input

```
var input = new string[] { "Apple", "Banana",
"Mango", "Apple", "Orange", "Mango",
"Strawberry", "Apple" };
```

Q: The problem is to write a query using LINQ and LAMBDA that will count the nunber of fruits.

ANS:

```
var input = new string[] { "Apple", "Banana",
"Mango", "Apple", "Orange", "Mango",
"Strawberry", "Apple" };

//Using LINQ

(from a in input

 group a by a into g

 select new

 {

   Key = g.Key

   ,Count = g.Count()

 })

 .ToList()

 .ForEach(i => Console.WriteLine("Number of
{0} is {1}", i.Key, i.Count));

//Using Lambda

 input

 .GroupBy(g => g)

 .Select(i => new { Key = i.Key, Count =
i.Count() })
 .ToList() .ForEach(i =>
Console.WriteLine("Number of {0} is {1}",
i.Key, i.Count));

Output
Number of Apple is 3
Number of Banana is 1
Number of Mango is 2
Number of Orange is 1
Number of Strawberry is 1
```

Q: How to assign a computed value to the same array back?
ANS:
Consider the below input

```
var input = new string[] { "Apple", "Banana",
"Mango", "Apple", "Orange", "Mango",
"Strawberry", "Apple" };
```

Q: The problem is to write a query using LINQ that will count the number of fruits and will assign back the value to the same array i.e. we should not create another array for storing the computed values.

ANS:

```
var input = new string[] { "Apple", "Banana",
"Mango", "Apple", "Orange", "Mango",
"Strawberry", "Apple" };

input = (from a in input

        group a by a into g

        where g.Count() >= 2

        select g.Key + " ( " + g.Count() + "
)").ToArray();
```

Output

```
Apple ( 3 )
Mango ( 2 )
```

Q: How will you obtain the length of the longest string in a Data table assuming that every column in of type string using Lambda?

ANS:

```
DataTable dt = new DataTable();

var maxLength =
dt.AsEnumerable().SelectMany(r =>
r.ItemArray.OfType<string>()).Max(i =>
i.Length);
```

First cast the Datatable to System.Collections.Generic.IEnumerable<T> object by using the AsEnumerable() extension method,then using the SelectMany extension method
to find out the items in the ItemArray whose type is string and finally figuring out the max length of the records.

Q: How will you obtain the length of the longest string in every column of a Datatable assuming that every column in of type string using Lambda?

ANS:

```
DataTable dt = new DataTable();

var maximumColumnsLengths =

Enumerable.Range(0,
dt.Columns.Count).Select(col =>
dt.AsEnumerable().Select(row =>
row[col]).OfType<string>().Max(val =>
val.Length)).ToList();
```

Q: Write a code snippet for Left outer join using LINQ?

ANS:

I failed to answer this question in my interview session. So I searched the answer for the question and found in MSDN. After my understanding I just wrote a simple program to understand the left outer join in LINQ.

```
namespace LeftOuterJoin

{    namespace LeftOuterJoin

   {

        class Employee

        {

            public string EmpName { get; set; }

            public int EmpAge { get; set; }

            public string EmpDeptID { get; set; }

        }

        class Department

        {

            public string DeptID { get; set; }

            public string DeptName { get; set; }

        }
```

```
    class Program

    {

        public static void Main(string[] args)

        {

            Employee objEmp1 = new Employee
{ EmpName = "Naga", EmpAge = 29,
EmpDeptID = "1" };

            Employee objEmp2 = new Employee
{ EmpName = "Sundar", EmpAge = 30,
EmpDeptID = "2" };

            Employee objEmp3 = new Employee
{ EmpName = "Siva", EmpAge = 28,
EmpDeptID = "3" };

            Employee objEmp4 = new Employee
{ EmpName = "Shankar", EmpAge = 31,
EmpDeptID = "4" };

            Department objDept1 = new
Department { DeptID = "1", DeptName = "IT"
};

            Department objDept2 = new
Department { DeptID = "2", DeptName =
"Admin" };

            Department objDept3 = new
Department { DeptID = "3", DeptName =
"Testing" };

            Department objDept4 = new
Department { DeptID = "4", DeptName = "HR"
};

            Department objDept5 = new
Department { DeptID = "5", DeptName =
"Sales" };

            //Creating List of Objects

            List<Employee> employees = new
List<Employee> { objEmp1, objEmp2,
objEmp3, objEmp4 };

            List<Department> depts = new
List<Department> { objDept1, objDept2,
objDept3, objDept4, objDept5 };

            //Left Side Department Right side
Employee

            var DeptEmpCollection = from dept
in depts

                                    join emp in
employees on dept.DeptID equals
emp.EmpDeptID into de

                                    from Employees in
de.DefaultIfEmpty()

                                    select new {
dept.DeptName, EmployeesName = (Employees
== null ? "--No Employee--" :
Employees.EmpName) };

            foreach (var EmpDept in
DeptEmpCollection)

            {

                Console.WriteLine("Dept {0},
EmpName {1}", EmpDept.DeptName,
EmpDept.EmployeesName);

            }

            Console.Read();

        }

    }

}
```

```
}
```

Q: Write a query to get the single employee name when there are many employees whose name is "test" in the database ?
ANS:

```
var employee = (from h in
contextobject.Employee

                where h.EmployeeName == "test"

                select
h).FirstOrDefault<Employee>();
```

Thanks and Regards
Akiii

Write a query to get the list of all employees whose name is "test" ?

```
var employeeList = (from h in context.Employee

                                        where
h.EmployeeName == "test"

                                        select
h).ToList<Employee>();
```

Q: IEnumerable Vrs IQueryable

ANS:
In Linq, to query data from database and collections we use IEnumerable and IQueryable. IEnumerable is inherited by IQueryable, so it has all features of it and of its capabilities. Both has their own importance to query data and data manipulation.

IEnumerable :-
1. It exists in System.Collection namespace.
2. It can move forward only over collection.
3. It is best to query data from in-memory collections like Array, List, etc.
4. It is suitable for Linq to Objects and Linq To Xml queries.
5. It doesn't support lazy loading, hence not suitable for paging like scenario.

```
DataContextClasses db= new
DataContextClasses();
IEnumerable<Employee>List
=dc.Employees.Where(m=>m.Name.StartsWith
("a"));
list=list.Take<Employee>(10);
```

IQueryable :-
1. It exists in System.Linq namespace.
2. It can move forward only over collection.
3. It is best to query data from out-memory like remote database.
4. It is suitable for Linq to Sql queries.
5. It support lazy loading, hence suitable for paging like scenario.

```
DataContextClasses db= new
DataContextClasses();
IQueryable<Employee>List
=dc.Employees.Where(m=>m.Name.StartsWith
("a"));
list=list.Take<Employee>(10);
```

Q: What is different between LINQ and Stored Procedure?
ANS:
1.We can debug LINQ as it is part of .Net, whereas Stored procedure can’t be.
2. Deployment is easy with LINQ as everything compiled into DLL whereas with Stored procedure script is required for deployment.
3. At compile time we can find error, if any in LINQ but it not possible with Stored procedure.

Q: Disadvantage of LINQ over Stored procedure?

ANS: Stored procedure compiles one time and executed every time whereas LINQ compiled everytime , so Stored Procedure is faster as compare to LINQ.

Q: What is var?
ANS:
Var is used to declare implicitly typed local variable means it tells compiler to figure out the type of the variable at compile time.
Since, 'var' is anonymous type, hence it is used where we don't know the type of output like joining of two tables.

```
var q = (from e in tblEmployee
join d in tblDepartment
on e.DeptID equals d.DeptID
select new
{
e.EmpID, e.FirstName, d.DeptName
});
```

Q: Write the basic steps to execute a LINQ query

ANS:
Obtain the data source (The data source can be either an SQL database or an XML file)
Create a query
Execute the query
All of the above
All Above

Q: What is the function of the DISTINCT clause in a LINQ query
ANS:
The DISTINCT clause returns the result set without the duplicate values

Q: What is DataContext?
ANS:
To communicate with a Database a connection must be made. In LINQ this connection is created by a DataContext.
Create connection to database.
It submits and retrieves object to database.
Converts objects to SQL queries and vice versa.

Q: What is a Lambda expression?
ANS: A Lambda expression is nothing but an Anonymous Function, can contain expressions and statements. Lambda expressions can be used mostly to create delegates or expression tree types. Lambda expression uses lambda operator => and read as 'goes to' operator.
Example: myExp = myExp/10;
The => operator has the same precedence as assignment (=) and is right-associative.

Q: Lambdas are used in method-based LINQ queries as arguments to standard query operator methods such as Where.
What are the four language extensions in C# 3.0 useful for LINQ
ANS:
a) Lambda Expressions--A lambda expression is an anonymous function that can be used to create delegates.
b) Anonymous Types--Anonymous types are used to create a set of read-only properties into a single object without defining a type first
c)Object Initializers--Object Initializers is a new offering in C# 3.0 to create and initialize the objects in one step.
d) Implicitly Typed Variables --Can be assigned to any type.keyword used is var

Q: What are the four LINQ Providers types in .NET Framework 3.0
ANS: The four LINQ Providers types introduced in .NET Framework 3.0
a. LINQ to DataSets - For handling LINQ queries against ADO.NET DataSets.
b. LINQ to Objects - For handling a LINQ query against a collection of objects
c. LINQ to XML - For handling an XPATH query against XML documents
d. LINQ to SQL - For handling LINQ queries against Microsoft SQL Server.

Q: What is the difference between OrderBy() and Sort() method over IList
ANS:

OrderBy() sorts and gives the view IEnumerable(). But underlying list is sorted or not changed.
Sort() modifies the underlying list.

Q: What is the difference between Select() and SelectMany()?
ANS:
Select() converts one type to another and returns enumerable.
SelectMany() flattens the elements and gives a combined list of elements.

Q: What is the difference between Skip() and SkipWhile() extension methods?
ANS:
Skip() will take an integer argument and skips the top n numbers from the given IEnumerable
SkipWhile() continues to skip the elements as long as the input condition is true. Once condition turns false it will return all remaining elements.

Q:What is the method to convert one type to another using Lambda expressions?
ANS:
Select() method

Q: What is Deferred Execution?
ANS:
Deferred execution is related to Lazy Evaluation.
In LINQ, the actual processing is deferred to a later stage until the elements are enumerated.
For eg: list.Where(i => i != 0) returns an object not the actual result.

Q: Is the method Where() using lambda expression returns values immediately.
ANS: Nope. It is deferred execution. It will return an object which contains the information to execute the query. The actual filtering happens when the enumeration of elements starts.

The enumeration could be started by using a foreach loop.

Q: Are Where() and TakeWhile() both having same functionalities?
ANS: No.
Where() will return all elements for given condition.
Take() will return elements until given condtion is true. Once condition turned false it exits the iteration and remaining elemments are not checked.
Q: Which is the Lambda Expression enabled extension method to check any elements satisfies given condition?
ANS:
Any()

Q: Which is the Lambda Expression enabled extension method to check all elements satisfies given condition?
ANS:
All()

Q: How to find the index of element using Where() with Lambda Expressions?
ANS: Use the two argumented Lambda method
Where((i, ix) => i == ix);

Q: How Where() method using Lambda Expression implemented for List in C#?
ANS: Using Extension Methods

Q: What is CAML?
ANS: Collaborative Application Markup Language (CAML) is the XML-based language that is used to build and customize Web sites
CAML can be used to do the following:
.Provide schema definition to the Web site provisioning system about how the site looks and acts.
.Define views and forms for data and page rendering or execution.
.Act as a rendering language that performs functions in the DLL like pulling a value from a particular field.

.Provide batch functionality for posting multiple
commands to the server using protocol.