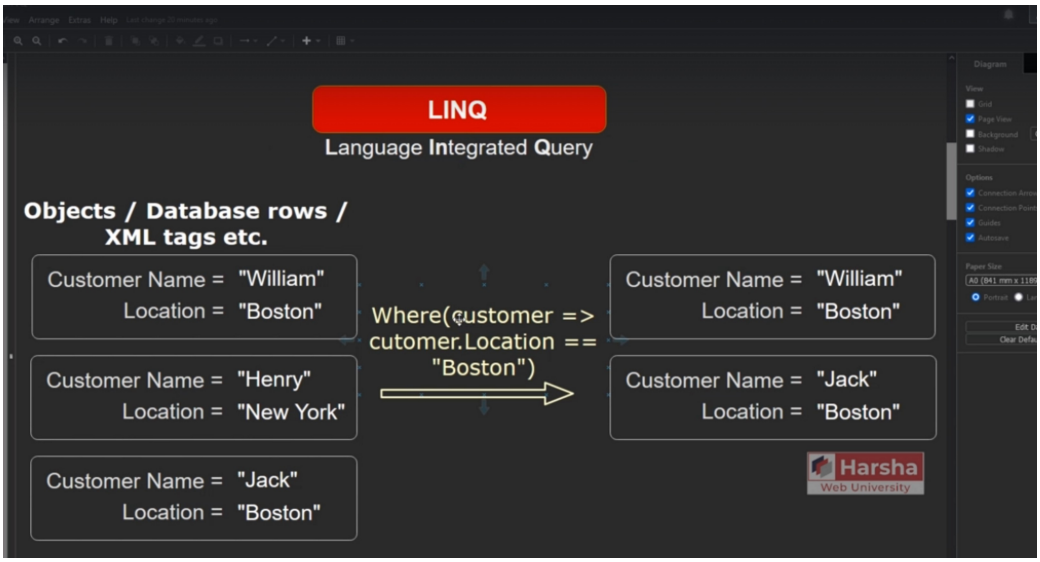
Section 25 : LINQ

28 July 2024 20:40

1. Linq Basics

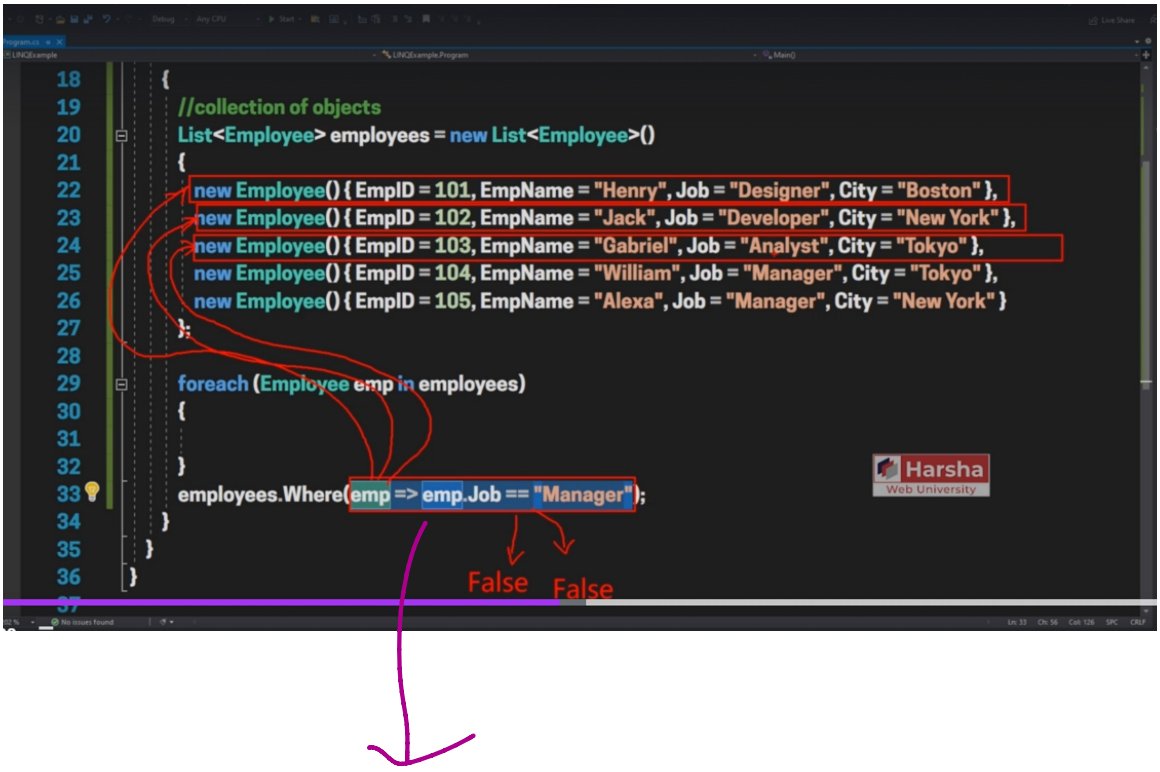




It says that you will write only one query against any type of data source your data can be in the form of list of objects or database rows maybe SQL server or oracle or any other database rows or you might have data in the form of other data sources such as xml tags or csv files etc

you will write only one type of queries against all types of data sources for example we are writing a query saying that I would like to retrieve the customers where the location equal to boston no matter what is the actual data source from this list of customers you will get the corresponding matching rows so you can print that results or you can store into another variable

or you can do anything with these results.

Link is actually a framework which has been supported by c-sharp and vb.net languages.



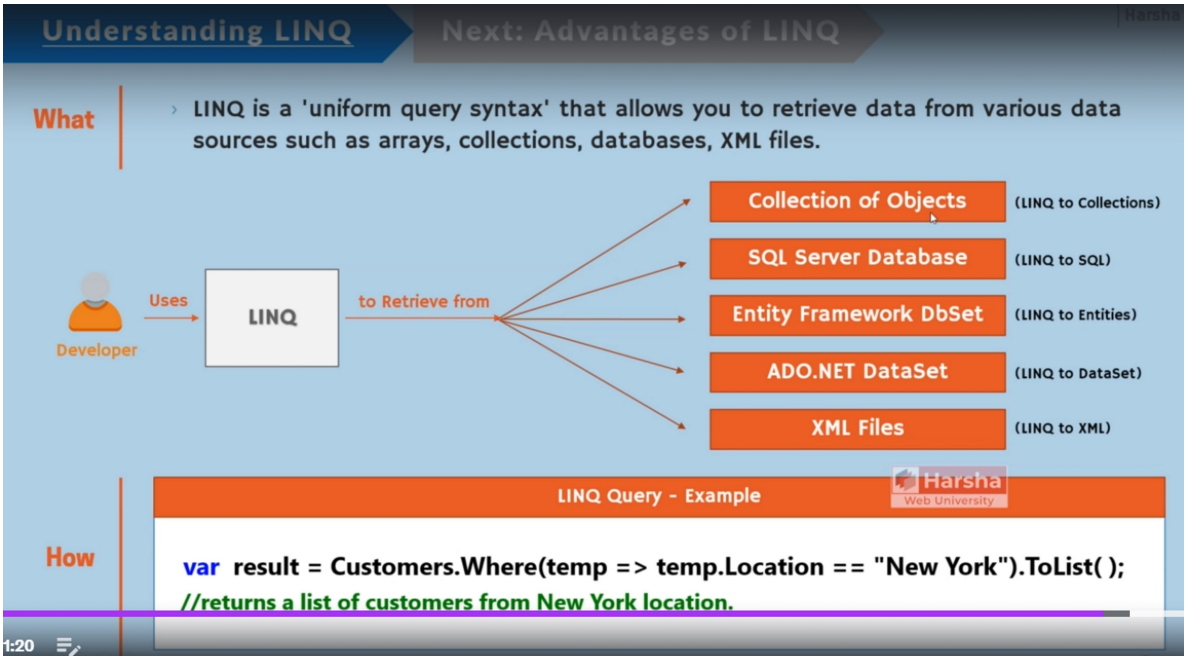

Lamda Expression

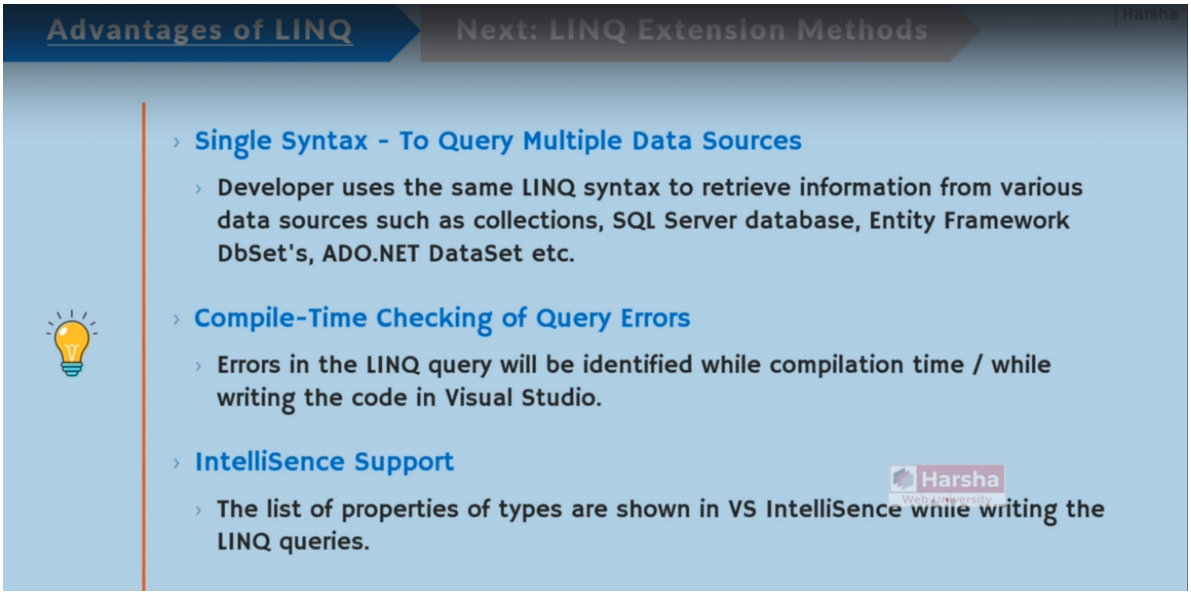
The group of yield return values can form a IEnumerable object.

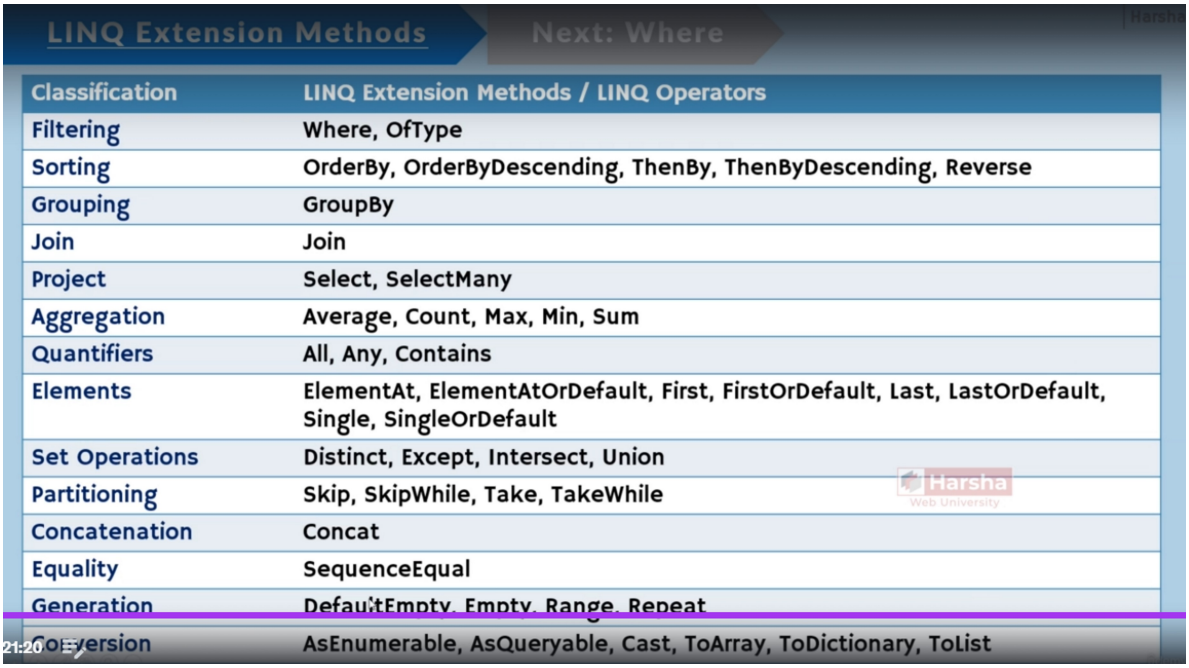
```
    new Employee() { EmpID = 103, EmpName = "Gabriel", Job = "Analyst", City = "Tokyo" },
    new Employee() { EmpID = 104, EmpName = "William", Job = "Manager", City = "Tokyo" },
    new Employee() { EmpID = 105, EmpName = "Alexa", Job = "Manager", City = "New York" }
};

foreach (Employee emp in employees)
{

}
var result = employees.Where(emp => emp.Job == "Manager");
```

IEnumerable<Employee>

Harsha Web University

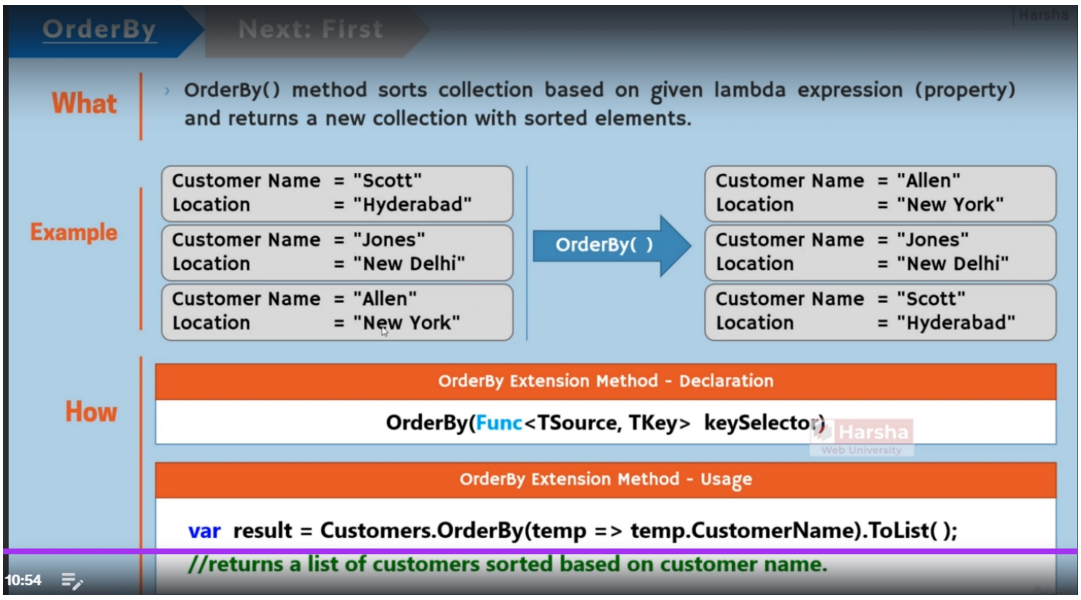


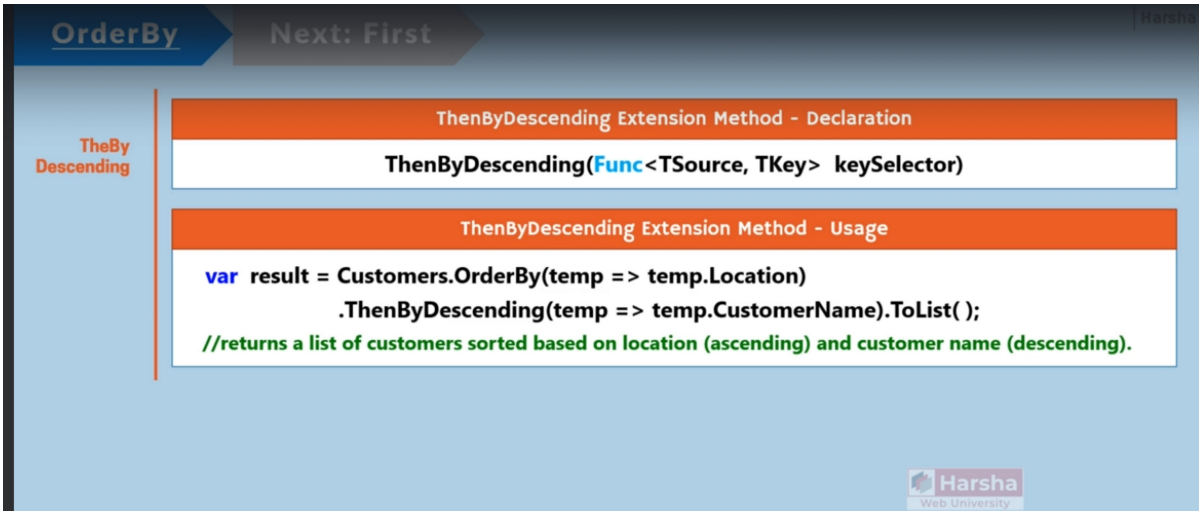

Advantages of LINQ
Next: LINQ Extension Methods
Harsha
Single Syntax - To Query Multiple Data Sources
Developer uses the same LINQ syntax to retrieve information from various data sources such as collections, SQL Server database, Entity Framework DbSet's, ADO.NET DataSet etc.
Compile-Time Checking of Query Errors
Errors in the LINQ query will be identified while compilation time / while writing the code in Visual Studio.
IntelliSence Support
The list of properties of types are shown in VS IntelliSence while writing the LINQ queries.
Harsha
Web University


LINQ Extension Methods
Next: Where
Harsha
Classification
LINQ Extension Methods / LINQ Operators
Filtering
Where, OfType
Sorting
OrderBy, OrderByDescending, ThenBy, ThenByDescending, Reverse
Grouping
GroupBy
Join
Join
Project
Select, SelectMany
Aggregation
Average, Count, Max, Min, Sum
Quantifiers
All, Any, Contains
Elements
ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault
Set Operations
Distinct, Except, Intersect, Union
Partitioning
Skip, SkipWhile, Take, TakeWhile
Harsha
Web University
Concatenation
Concat
Equality
SequenceEqual
Generation
DefaultEmpty, Empty, Range, Repeat
Conversion
AsEnumerable, AsQueryable, Cast, ToArray, ToDictionary, ToList
21:20

2. OrderBy

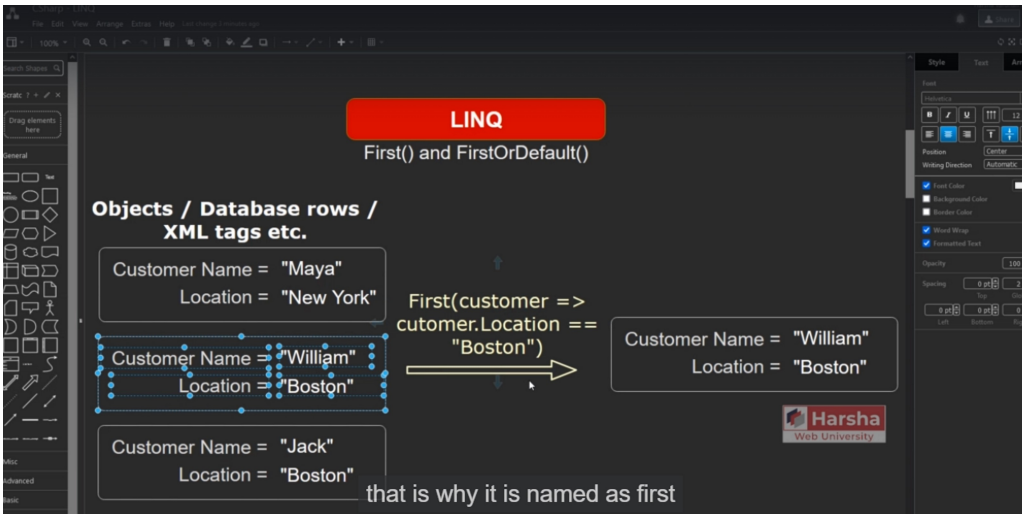
# OrderBy

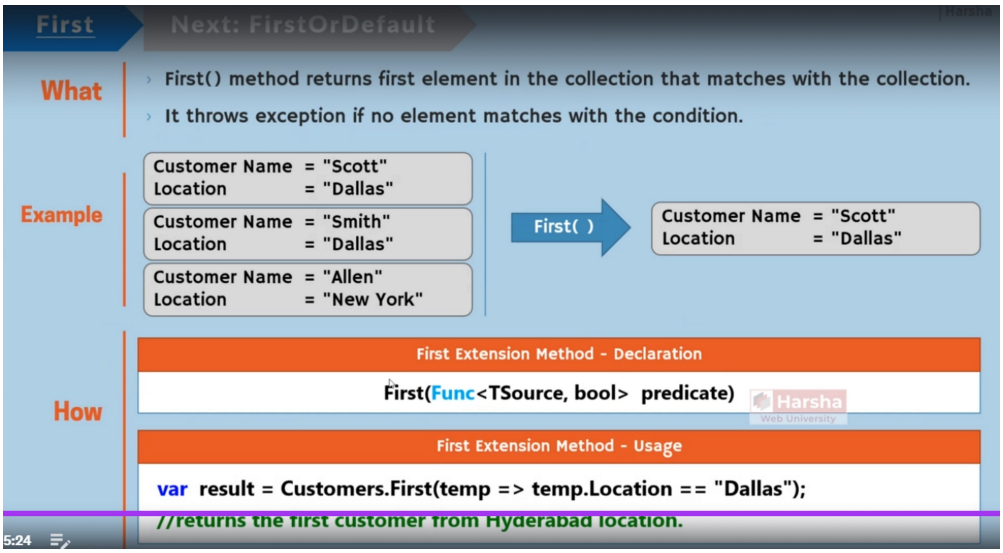


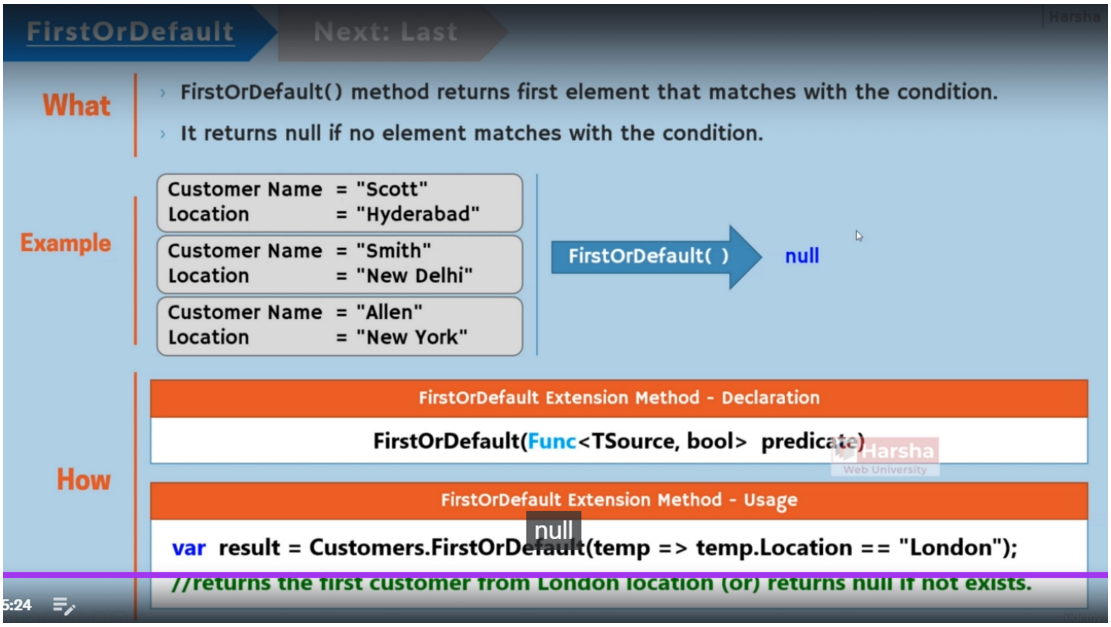




3. FirstOrFirstDefault

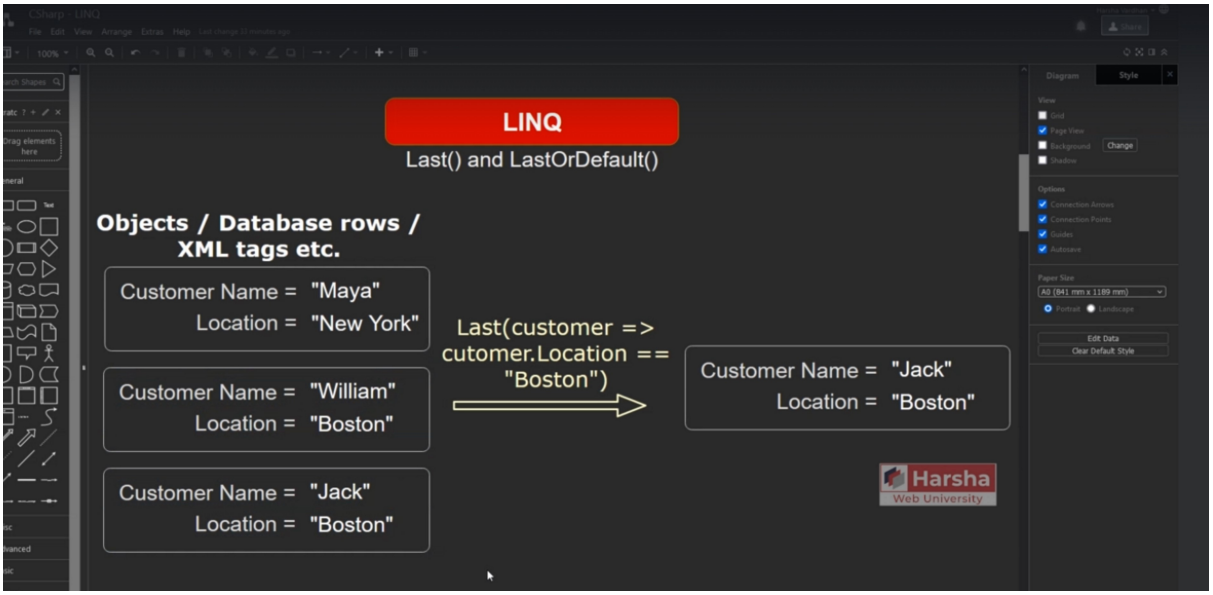
# First() and FirstOrDefault()

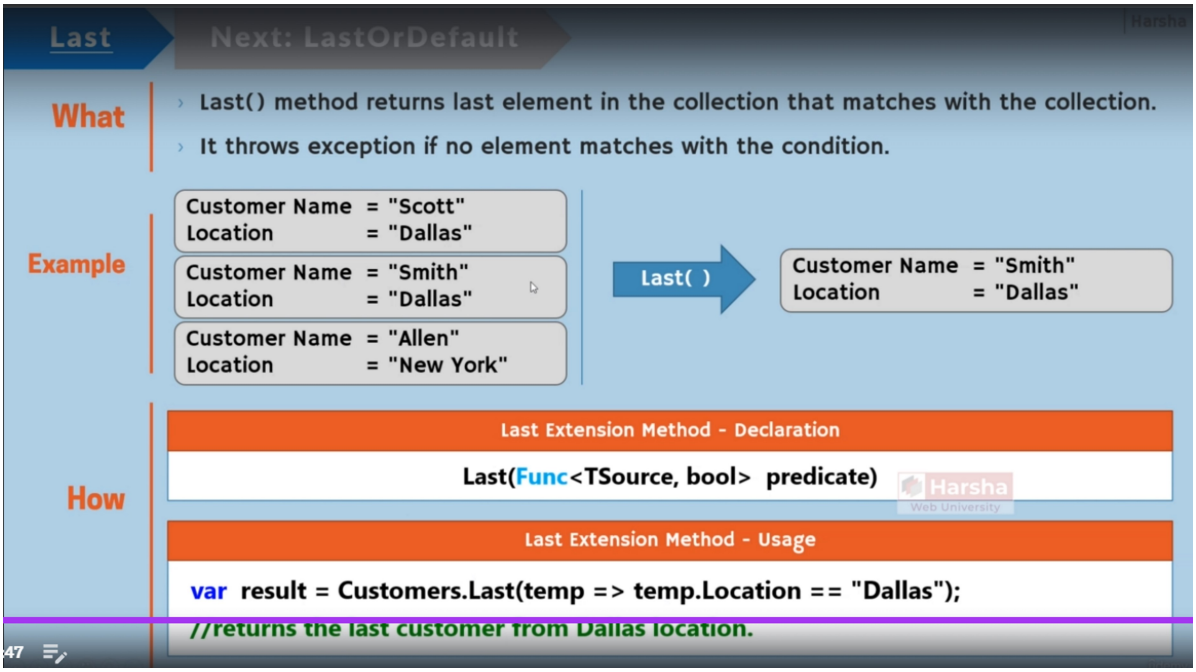


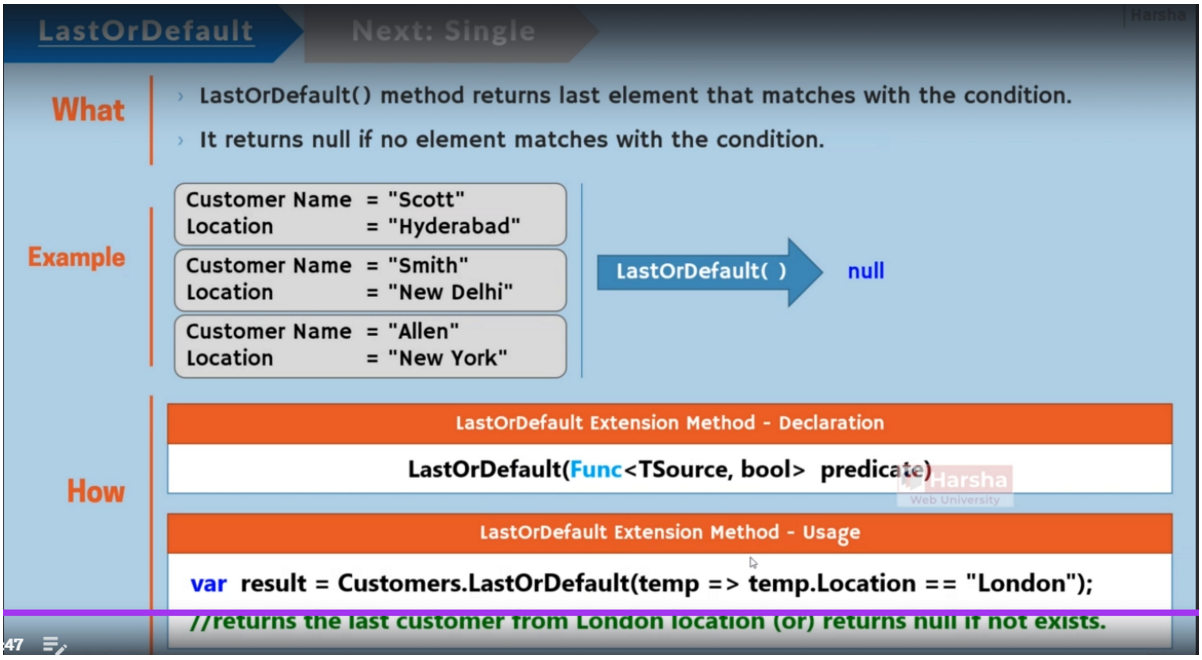





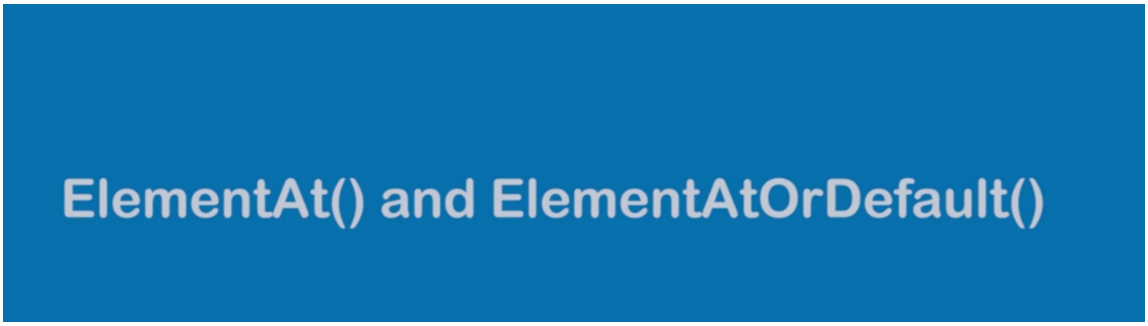

4. Last and LastOrDefault

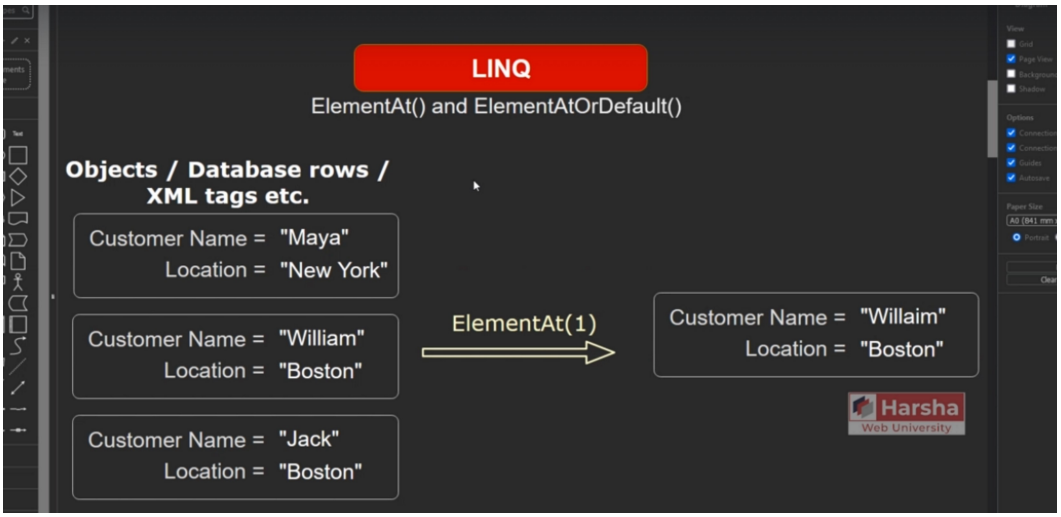

LINQ
Last() and LastOrDefault()
Objects / Database rows / XML tags etc.
Customer Name = "Maya"
Location = "New York"
Customer Name = "William"
Location = "Boston"
Customer Name = "Jack"
Location = "Boston"
Last(customer => cutomer.Location == "Boston")
Customer Name = "Jack"
Location = "Boston"
Harsha
Web University

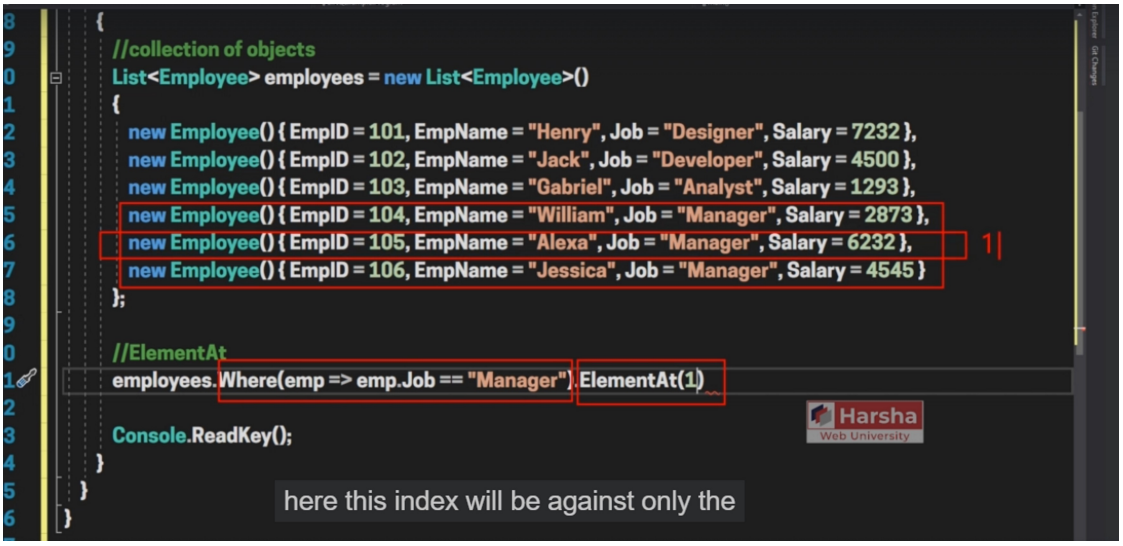

Last
Next: LastOrDefault
What
Last() method returns last element in the collection that matches with the collection.
It throws exception if no element matches with the condition.
Example
Customer Name = "Scott"
Location = "Dallas"
Customer Name = "Smith"
Location = "Dallas"
Customer Name = "Allen"
Location = "New York"
Last( )
Customer Name = "Smith"
Location = "Dallas"
How
Last Extension Method - Declaration
Last(Func<TSource, bool> predicate)
Harsha
Web University
Last Extension Method - Usage
var result = Customers.Last(temp => temp.Location == "Dallas");
//returns the last customer from Dallas location.



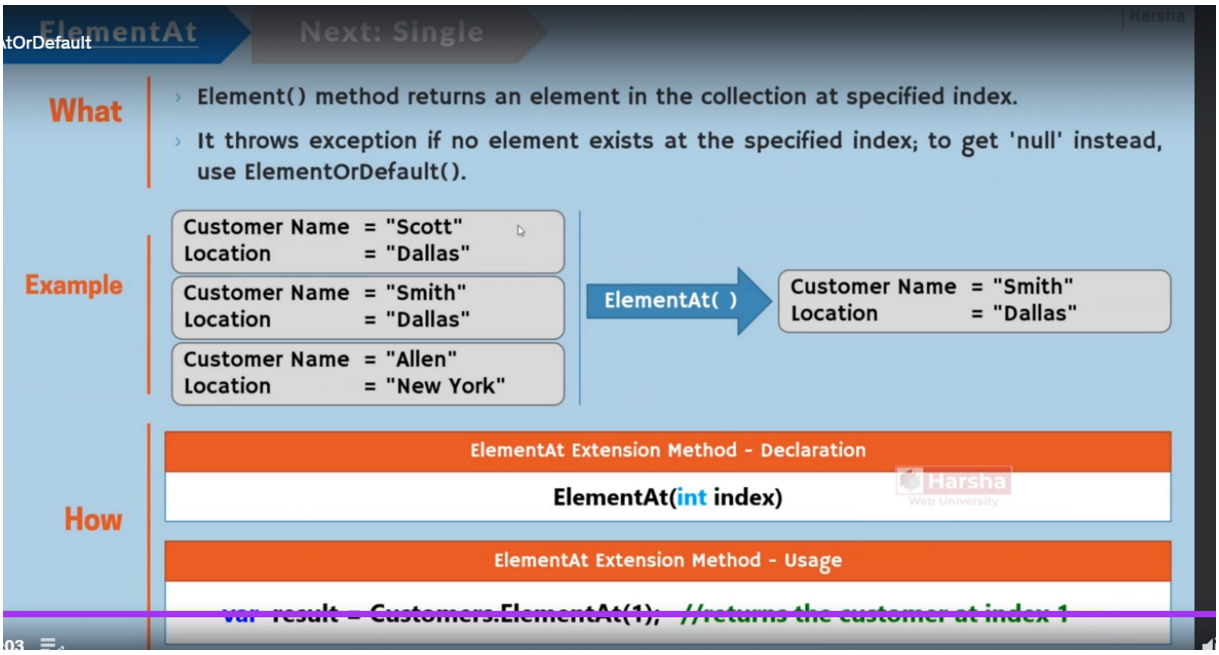

5. ElementAt() and ElementAtOrDefault()



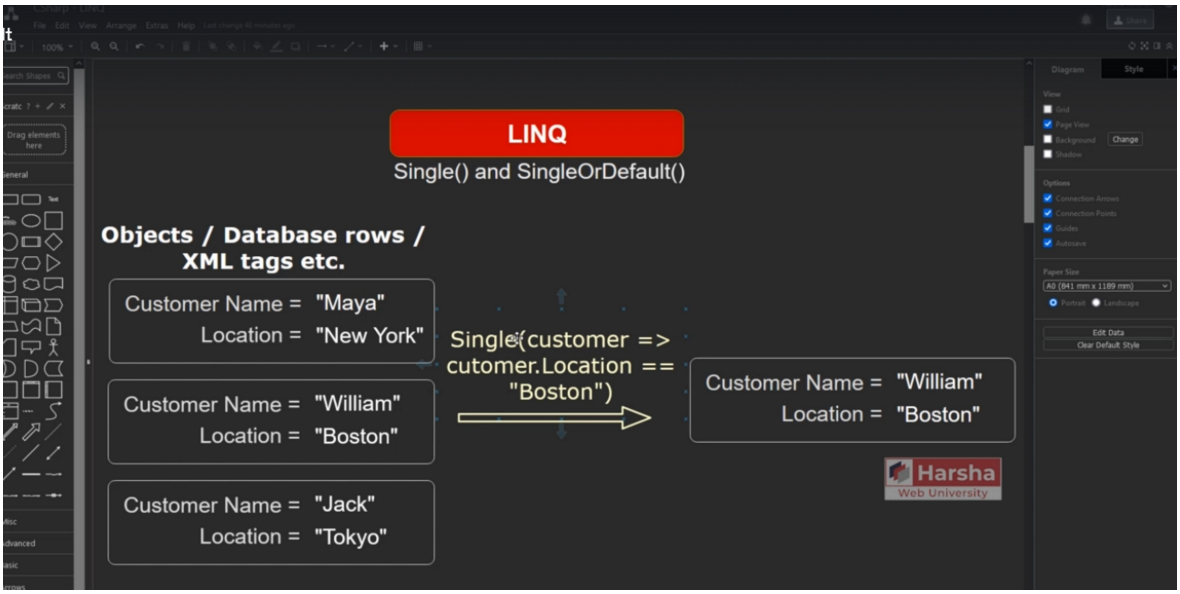






Where() methods return a list of Ienumerable, based on that ElementAt method executes for each of the employee object.



# Single() and SingleOrDefault()




It's similar to First /FirstOrDefulat. The difference is, Single method expects there must be only one matching element in the whole list based on the given condition.

***Single*** *method throws exception in these following cases:*

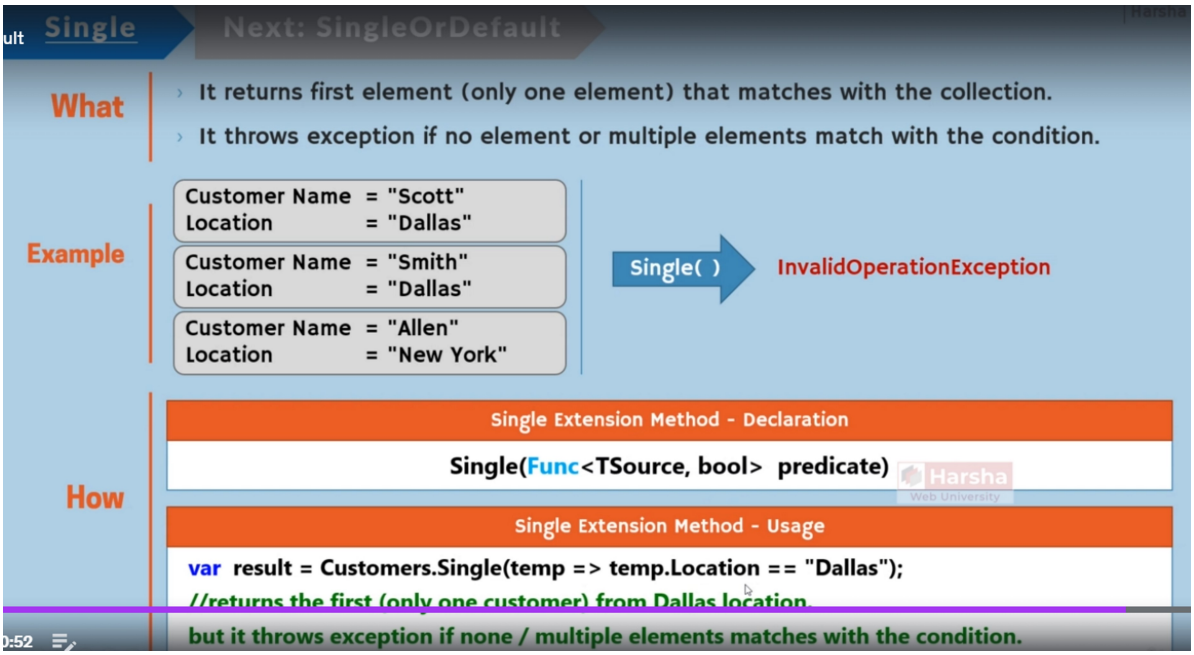
i. If more than one matching element is found, it throws exception.

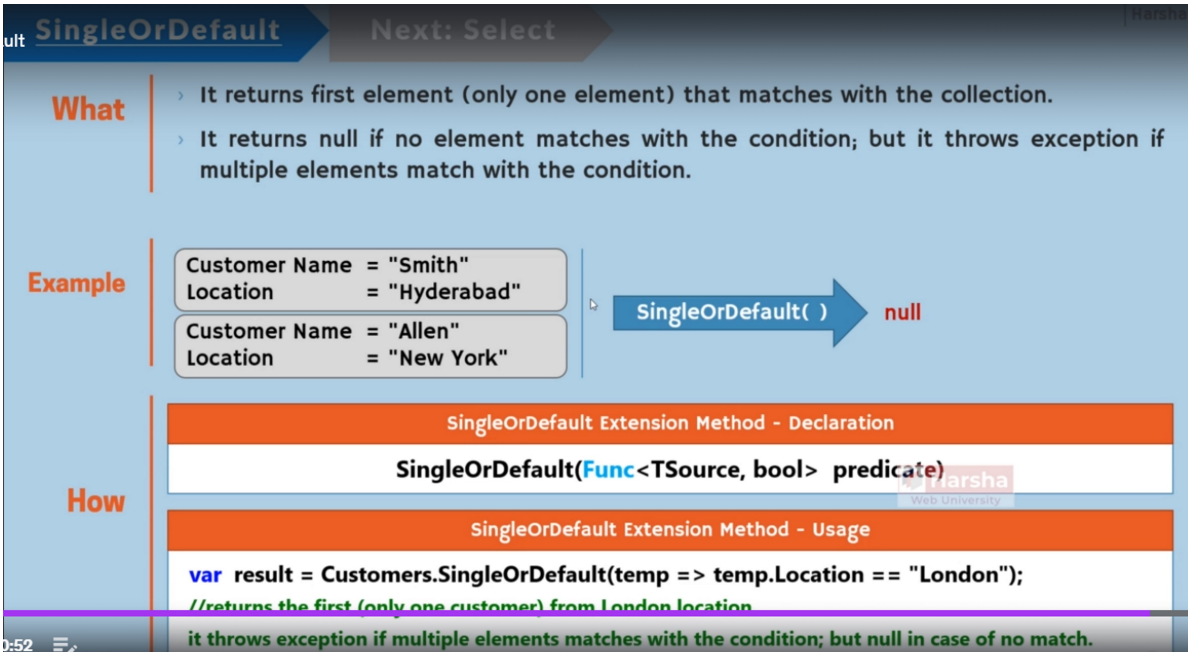
ii. it no elements is found with the given condition, it throws exception.

***SingleOrDefault*** *method throws exception in these following cases:*

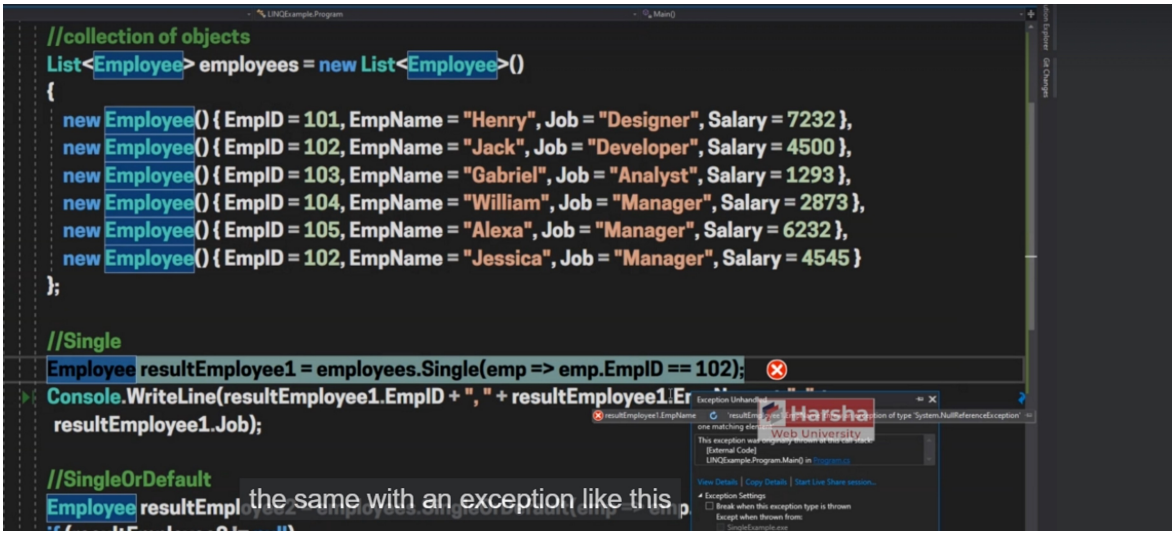
i. If more than one matching element is found, it throws exception.

**it returns Null if no element is found rather than throwing exception.**



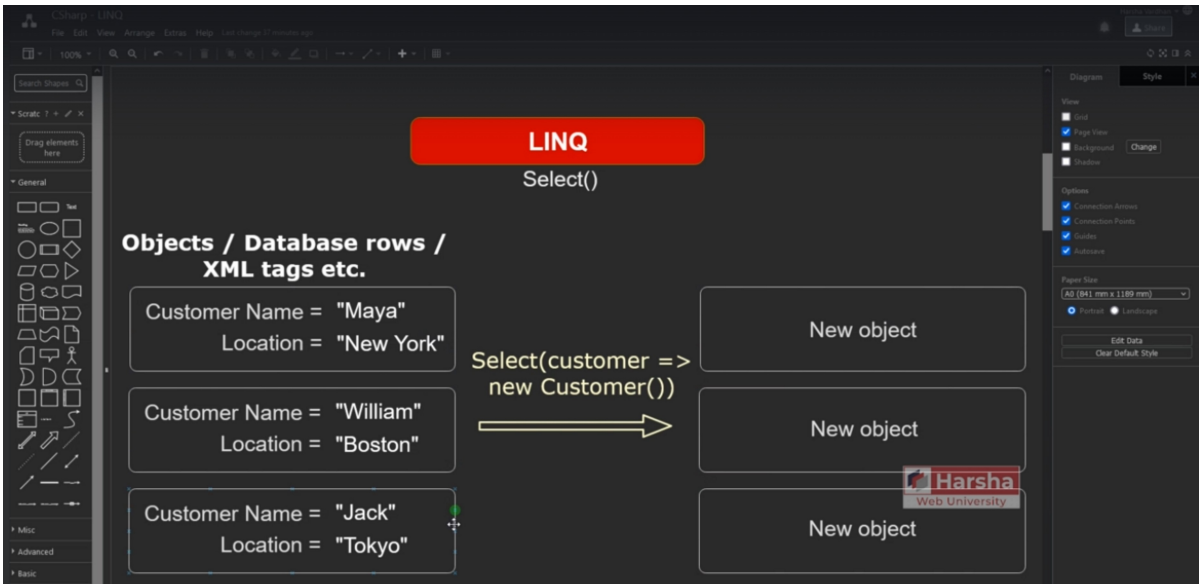


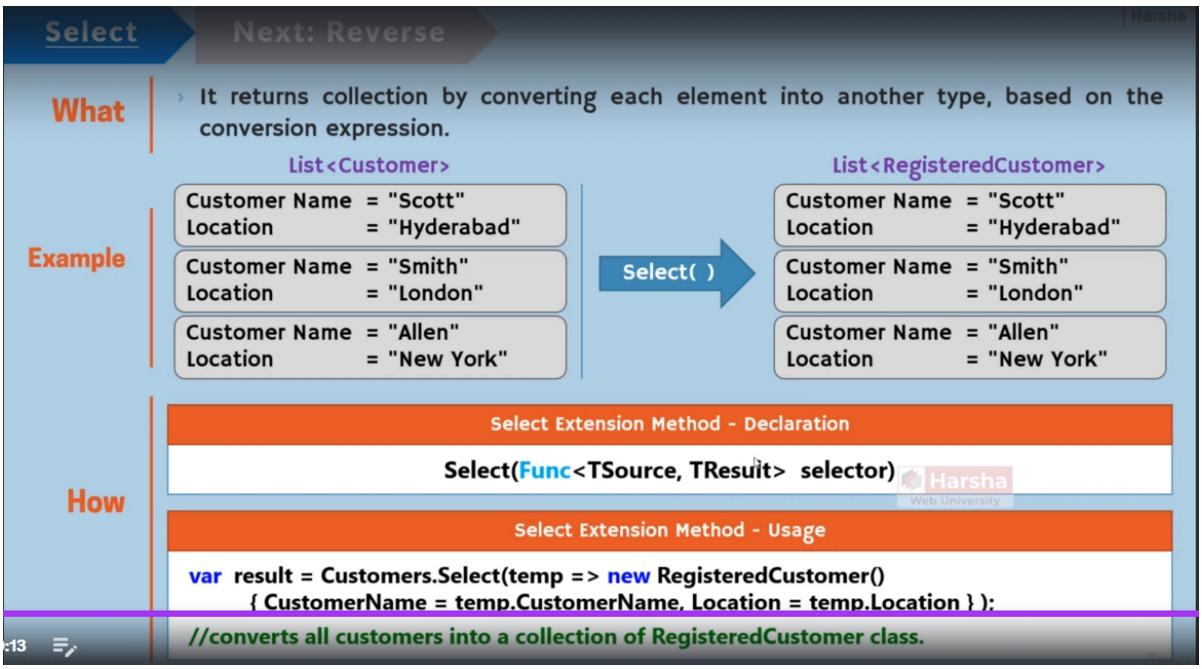

for this case single method is useful, where EmpId is duplicate.



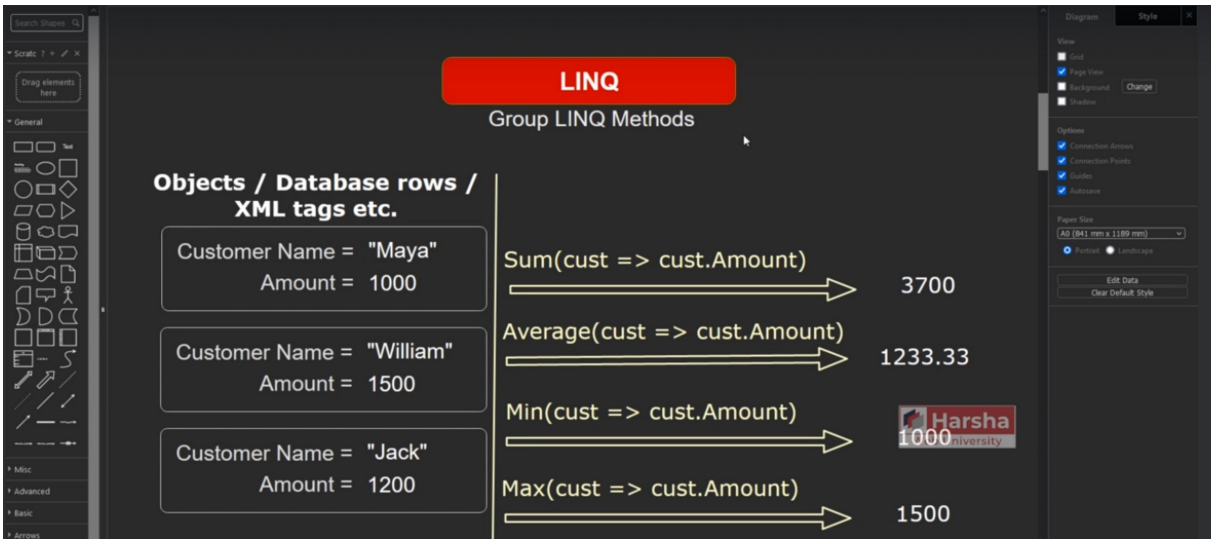

Select() LINQ Method

# Select()


LINQ
Select()
Objects / Database rows / XML tags etc.
Customer Name = "Maya"
Location = "New York"
Customer Name = "William"
Location = "Boston"
Customer Name = "Jack"
Location = "Tokyo"
Select(customer => new Customer())
New object
New object
New object
Harsha
Web University


Select
Next: Reverse
What
It returns collection by converting each element into another type, based on the conversion expression.
Example
List<Customer>
Customer Name = "Scott"
Location = "Hyderabad"
Customer Name = "Smith"
Location = "London"
Customer Name = "Allen"
Location = "New York"
Select( )
List<RegisteredCustomer>
Customer Name = "Scott"
Location = "Hyderabad"
Customer Name = "Smith"
Location = "London"
Customer Name = "Allen"
Location = "New York"
How
Select Extension Method - Declaration
Select(Func<TSource, TResult> selector)
Select Extension Method - Usage
var result = Customers.Select(temp => new RegisteredCustomer()
{ CustomerName = temp.CustomerName, Location = temp.Location } );
//converts all customers into a collection of RegisteredCustomer class.
Harsha
Web University

# Min(), Max(), Count(), Sum(), Average()




| Classification | LINQ Extension Methods / LINQ Operators |
|---|---|
| Filtering | Where, OfType |
| Sorting | OrderBy, OrderByDescending, ThenBy, ThenByDescending, Reverse |
| Grouping | GroupBy |
| Join | Join |
| Project | Select, SelectMany |
| Aggregation | Average, Count, Max, Min, Sum |
| Quantifiers | All, Any, Contains |
| Elements | ElementAt, ElementAtOrDefault, First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault |
| Set Operations | Distinct, Except, Intersect, Union |
| Partitioning | Skip, SkipWhile, Take, TakeWhile |
| Concatenation | Concat |
| Equality | SequenceEqual |
| Generation | DefaultEmpty, Empty, Range, Repeat |
| Conversion | AsEnumerable, AsQueryable, Cast, ToArray, ToDictionary, ToList |