Advance Programming Techniques (APT)

Lecture # 9

Ehtisham Rasheed

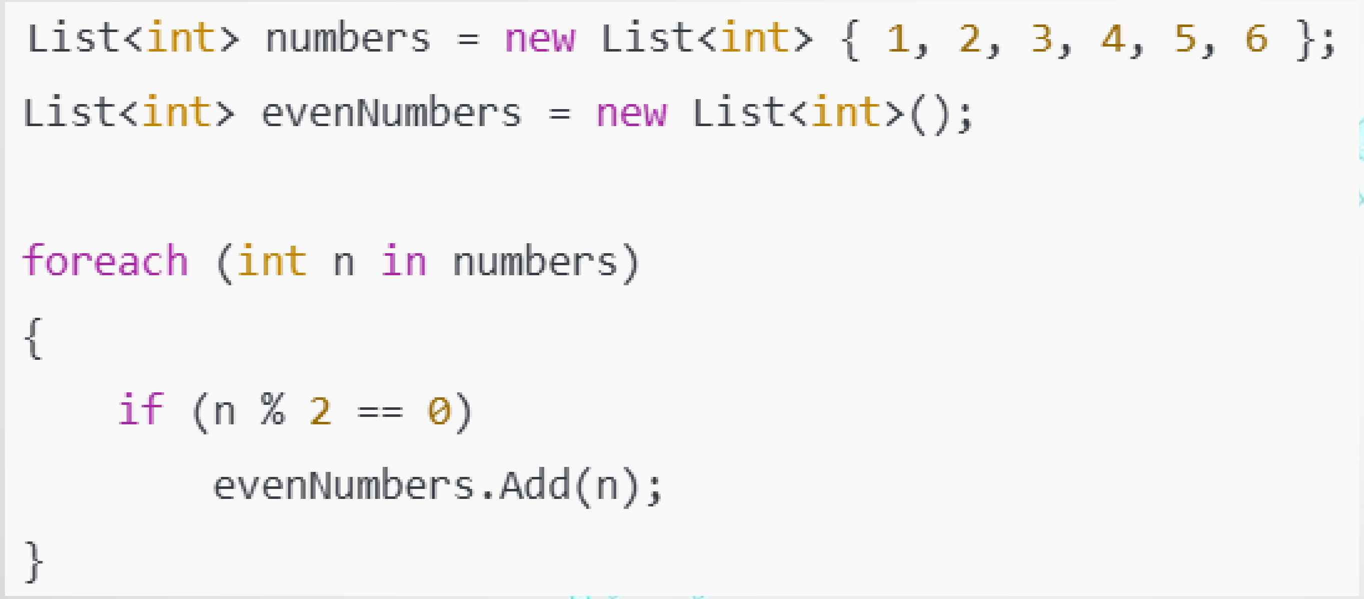
Department of Computer Science
University of Gurjat, Gujrat

LINQ

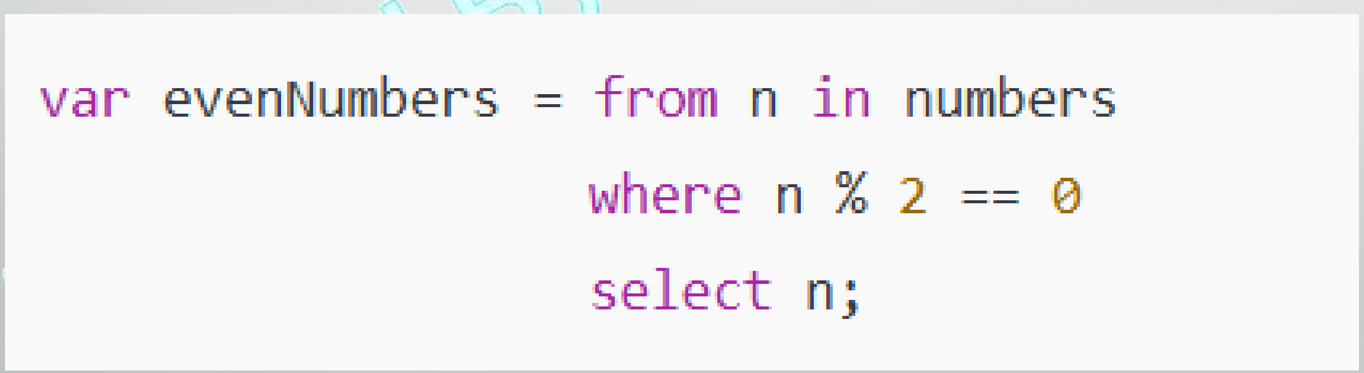
- LINQ stands for Language Integrated Query
- Offers easy data access from objects, databases, XML and many more
- Instead of writing loops and filters manually, we can use LINQ
- Key Benefits:
  - Consistent query syntax across different data sources
  - Type safety and IntelliSense support
  - Better readability and maintainability

Prepared By: Ehtisham Rasheed

Before and After LINQ



```
List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6 };
List<int> evenNumbers = new List<int>();

foreach (int n in numbers)
{
    if (n % 2 == 0)
        evenNumbers.Add(n);
}
```



```
var evenNumbers = from n in numbers
                  where n % 2 == 0
                  select n;
```

sheed
BY:
P

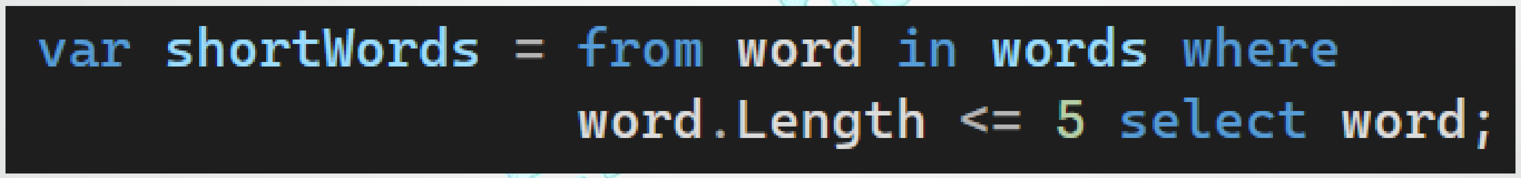
LINQ Example

```csharp
string[] words = {"hello", "wonderful", "LINQ", "beautiful", "world"};

//Get only short words
var shortWords = from word in words where word.Length <= 5 select word;

//Print each word out
foreach (var word in shortWords) {
   Console.WriteLine(word);
}
```

Prepa sheed

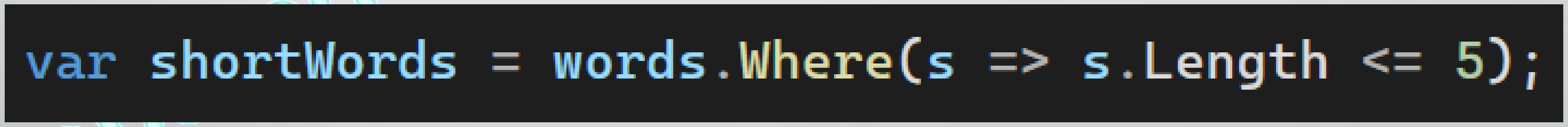
Syntax of LINQ

- There are two syntaxes of LINQ
- Query (Comprehension) Syntax



```
var shortWords = from word in words where
                 word.Length <= 5 select word;
```

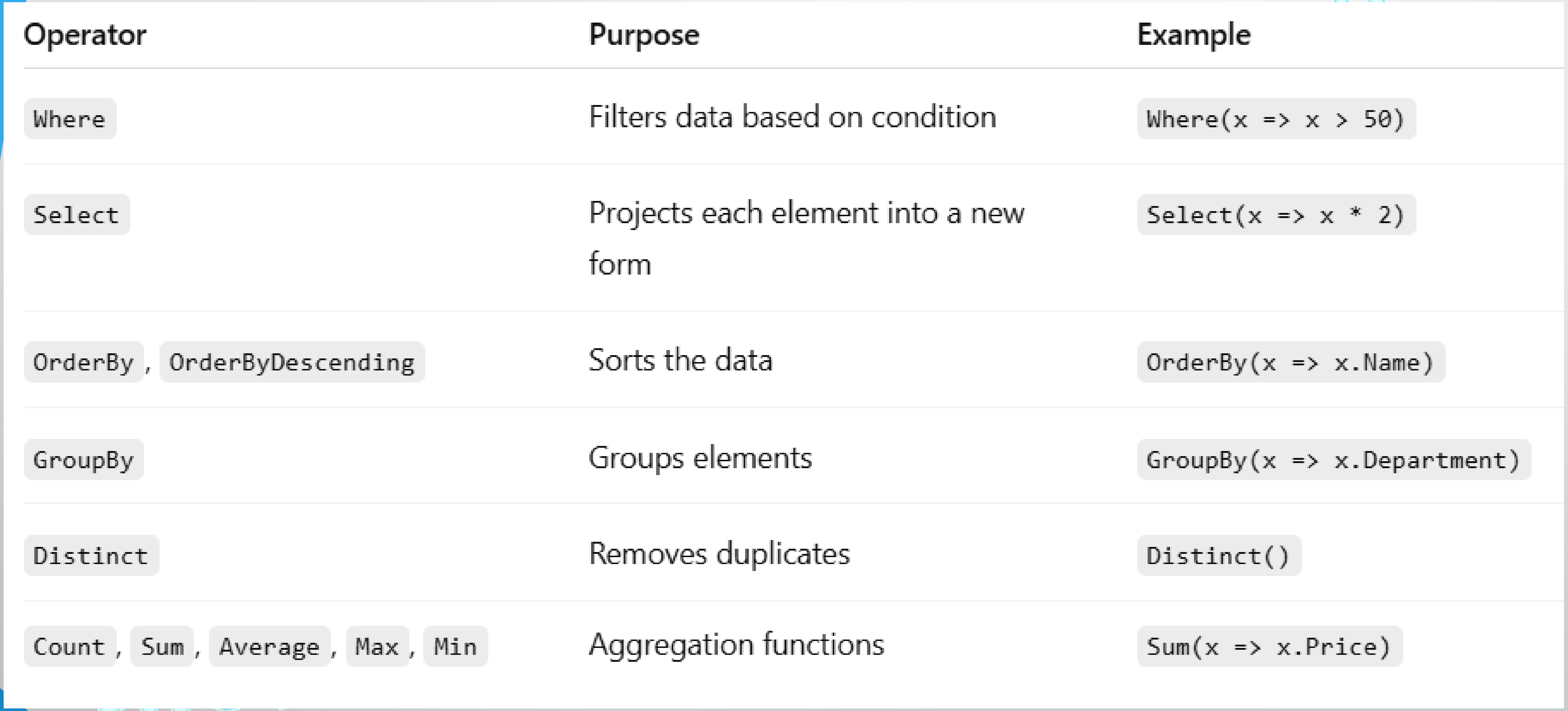
- Lamda (Method) Syntax



```
var shortWords = words.Where(s => s.Length <= 5);
```

Prep[illegible]d By: [illegible]ham Rasheed

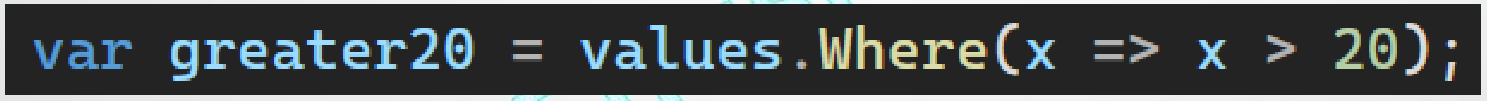
Basic LINQ Operators



Operator	Purpose	Example
`Where`	Filters data based on condition	`Where(x => x > 50)`
`Select`	Projects each element into a new form	`Select(x => x * 2)`
`OrderBy`, `OrderByDescending`	Sorts the data	`OrderBy(x => x.Name)`
`GroupBy`	Groups elements	`GroupBy(x => x.Department)`
`Distinct`	Removes duplicates	`Distinct()`
`Count`, `Sum`, `Average`, `Max`, `Min`	Aggregation functions	`Sum(x => x.Price)`

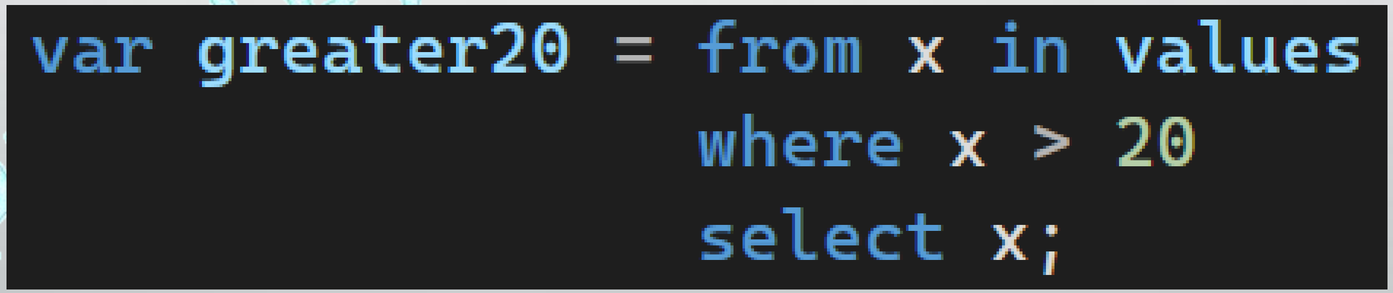
Where – Filtering Elements

- Filters a sequence (like a list or array) based on a condition
- It returns only those elements that satisfy the condition
- Method Syntax



```
var greater20 = values.Where(x => x > 20);
```

- Query Syntax

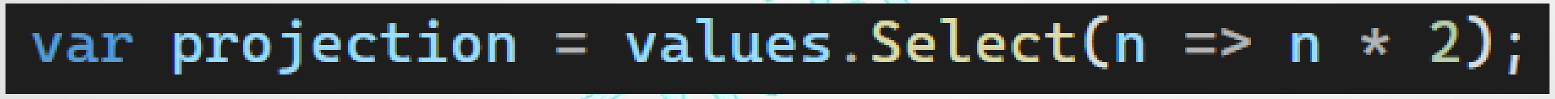


```
var greater20 = from x in values
                where x > 20
                select x;
```

Prepared By: Ehtisham Rasheed

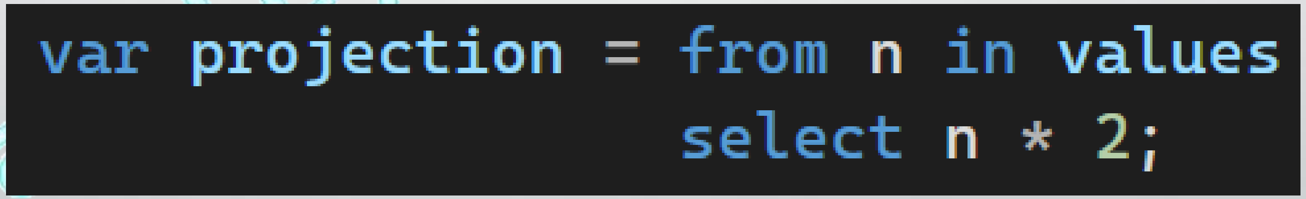
Select – Projection / Transfomation

- Transforms elements of a sequence into a new form
- Used to select one or more fields or perform calculations
- Method Syntax



```
var projection = values.Select(n => n * 2);
```

- Query Syntax

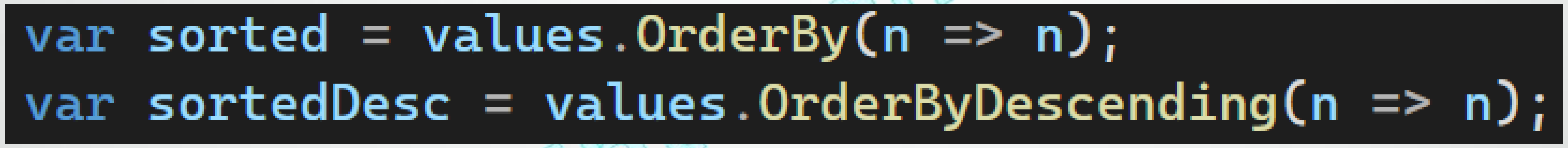


```
var projection = from n in values
                 select n * 2;
```

Prepared By: Ehtisham Rasheed

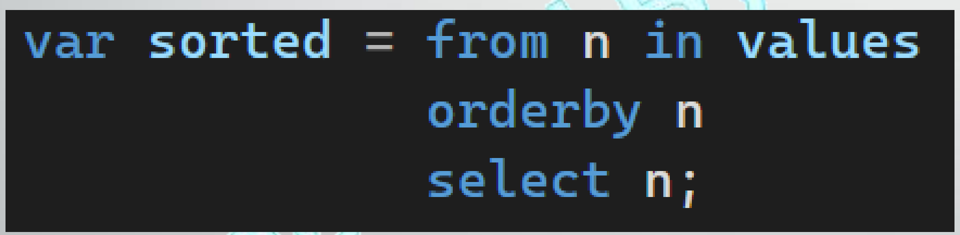
OrderBy / OrderByDescending – Sorting

- Sorts the elements in ascending or descending order
- Method Syntax

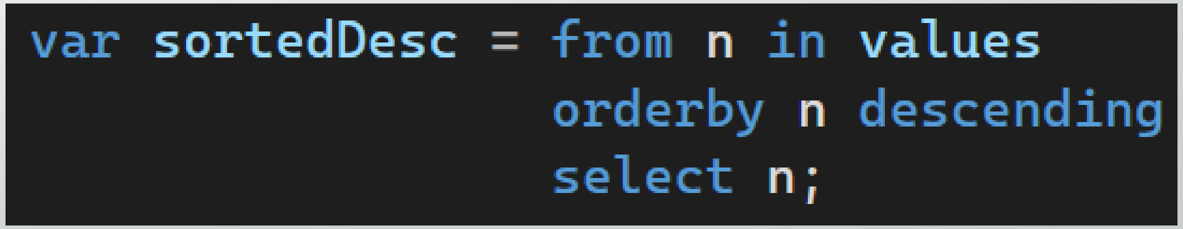


```
var sorted = values.OrderBy(n => n);
var sortedDesc = values.OrderByDescending(n => n);
```

- Query Syntax



```
var sorted = from n in values
             orderby n
             select n;
```

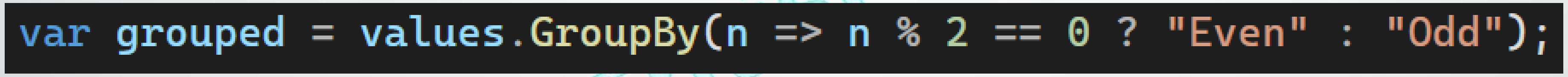


```
var sortedDesc = from n in values
                 orderby n descending
                 select n;
```

Prep By: Eh m Rasheed

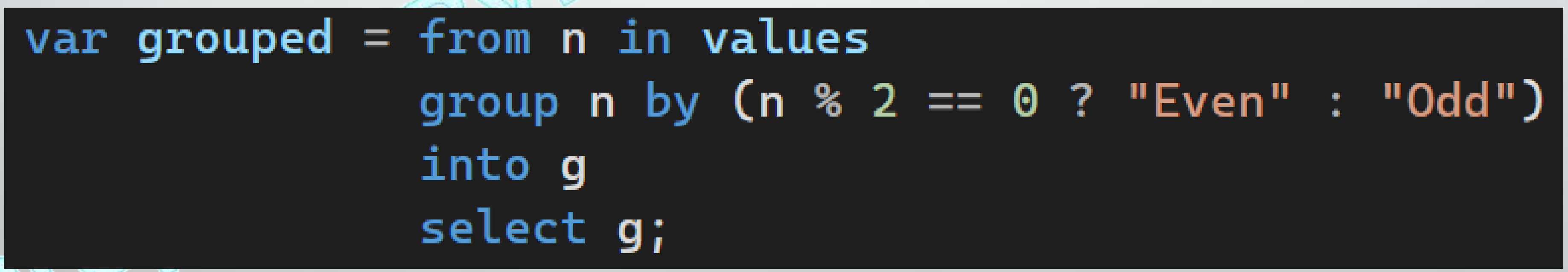
GroupBy – Grouping Elements

- Groups elements based on a key
- Each group is represented as an IGrouping<TKey, TElement>
- Method Syntax



```
var grouped = values.GroupBy(n => n % 2 == 0 ? "Even" : "Odd");
```

- Query Syntax

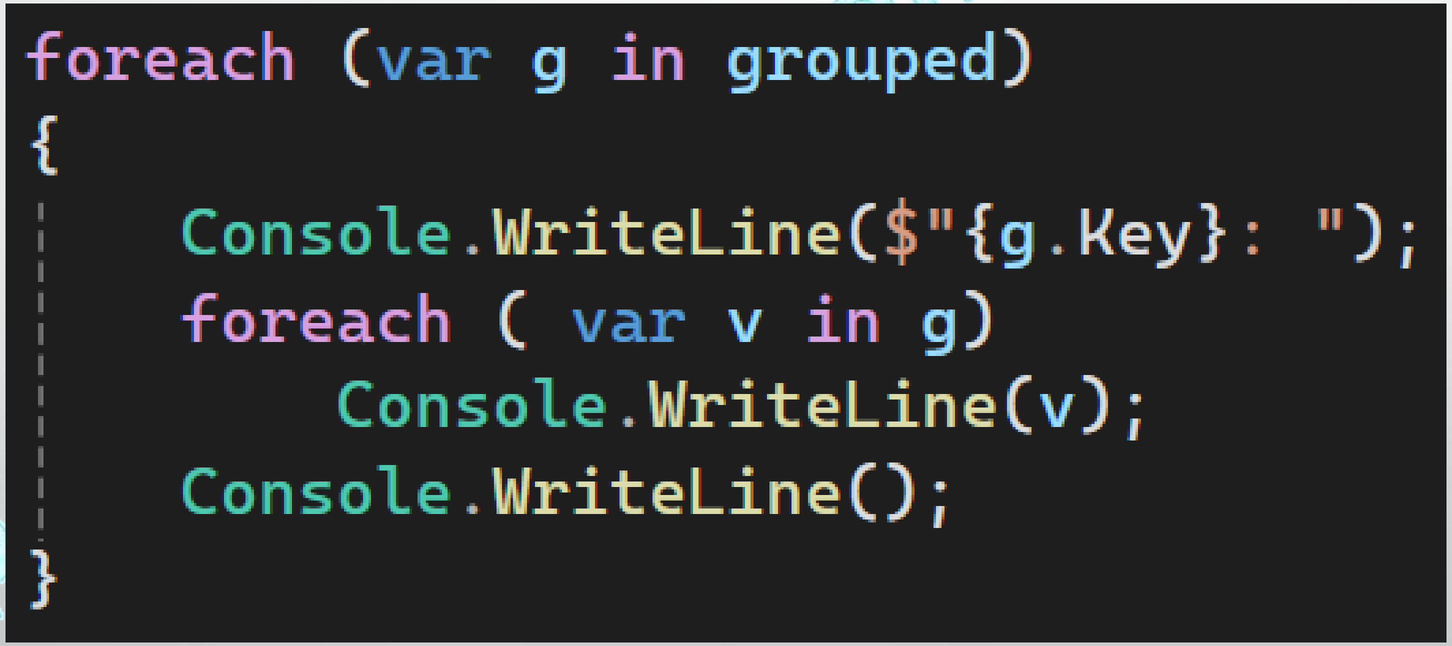


```
var grouped = from n in values
              group n by (n % 2 == 0 ? "Even" : "Odd")
              into g
              select g;
```

Prof. By: Ehtisham Rasheed

GroupBy – Grouping Elements

- You can loop over groups

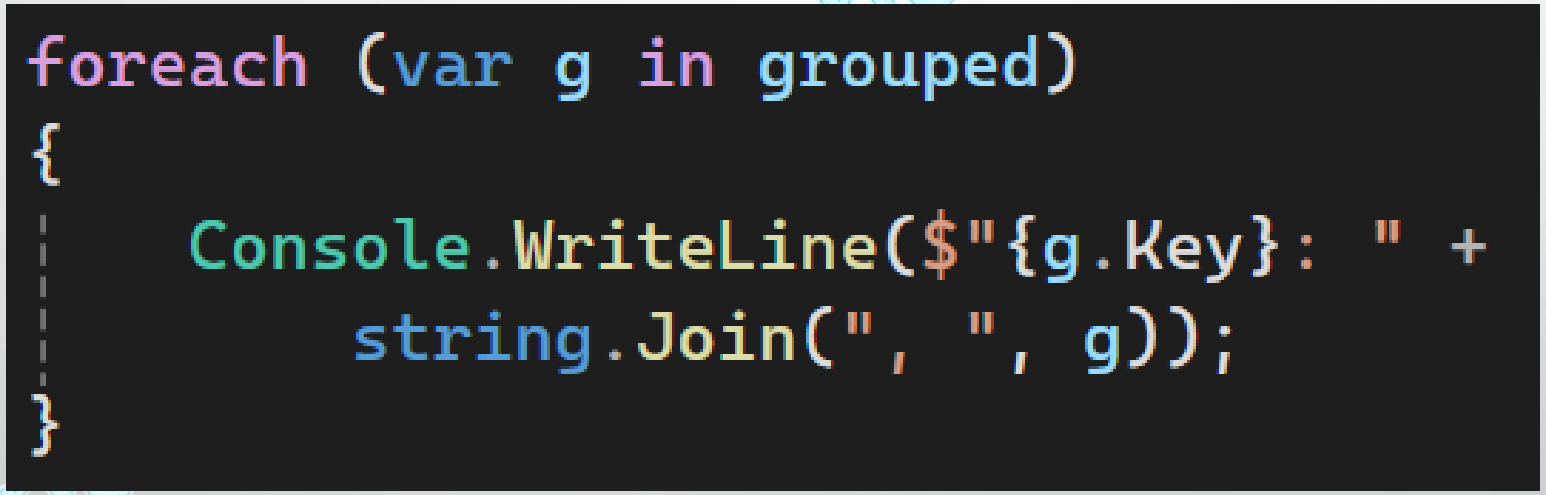


```
foreach (var g in grouped)
{
    Console.WriteLine($"{g.Key}: ");
    foreach ( var v in g)
        Console.WriteLine(v);
    Console.WriteLine();
}
```

m Rasheed
Prep

GroupBy – Grouping Elements

- You can loop over groups

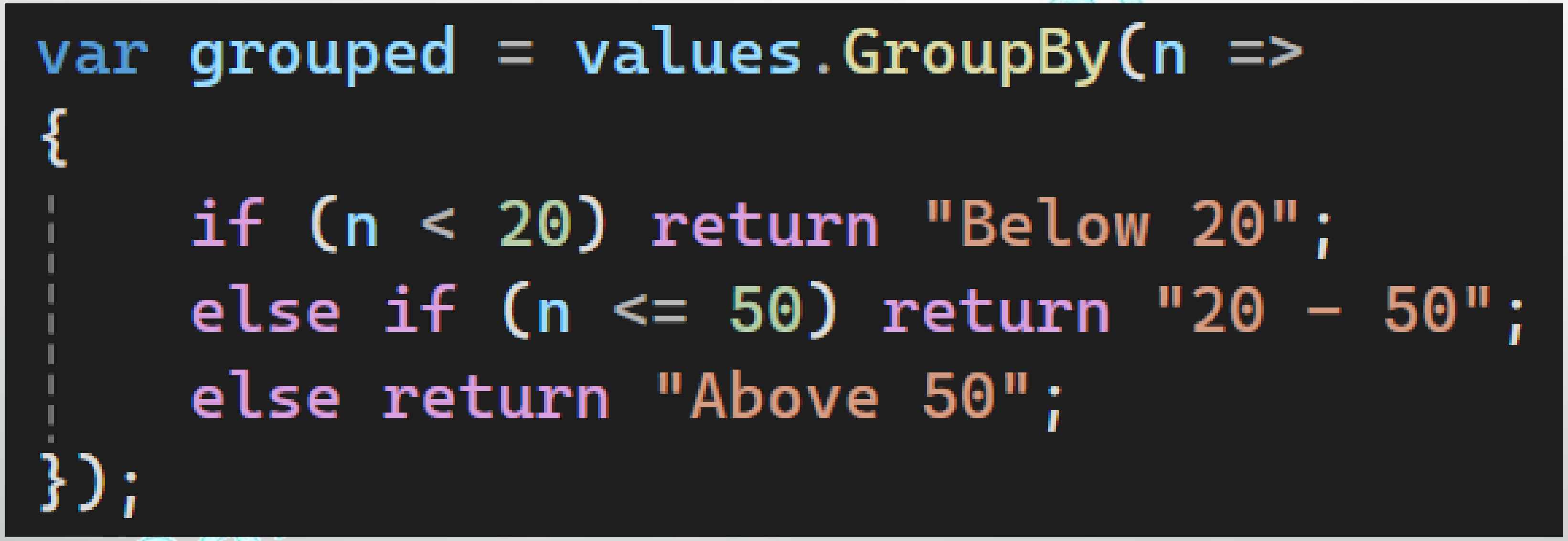


```
foreach (var g in grouped)
{
    Console.WriteLine($"{g.Key}: " +
        string.Join(", ", g));
}
```

Prep... ...ham Rasheed

GroupBy – More Than Two Groups

- Lamda Syntax

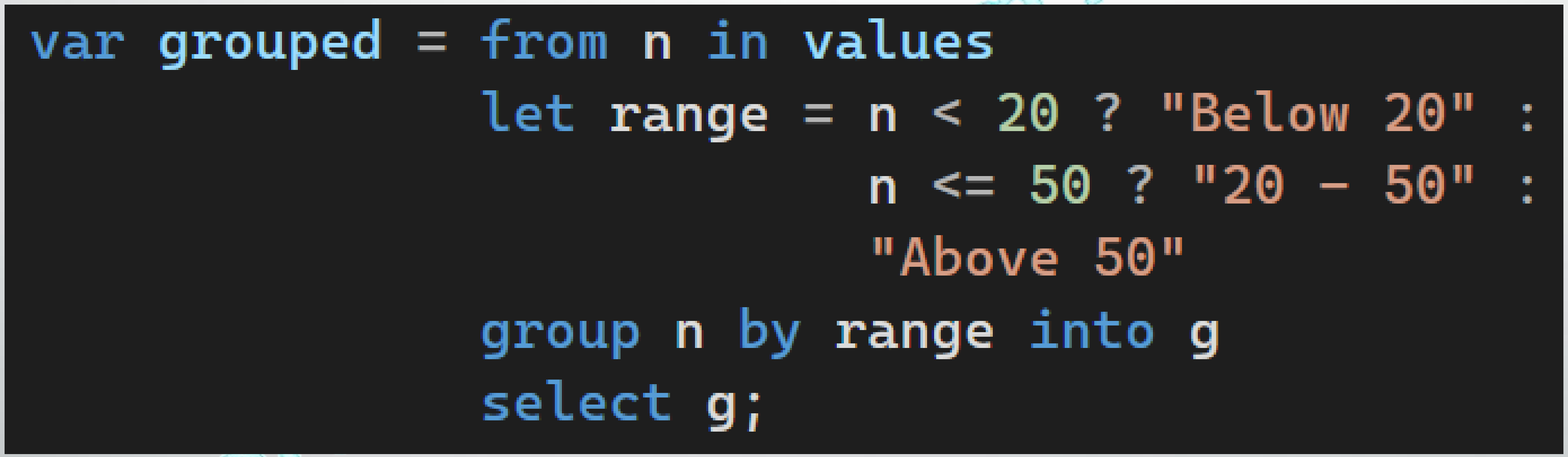


```
var grouped = values.GroupBy(n =>
{
    if (n < 20) return "Below 20";
    else if (n <= 50) return "20 - 50";
    else return "Above 50";
});
```

Prepa
m Rasheed

GroupBy – More Than Two Groups

- Query Syntax



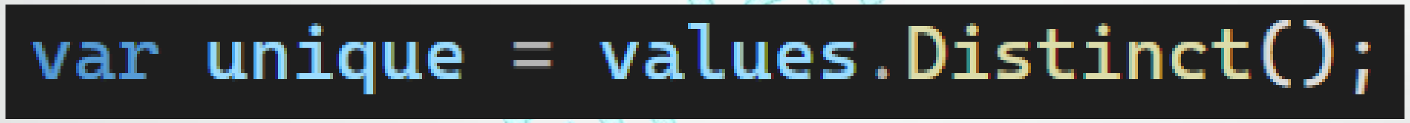
```
var grouped = from n in values
              let range = n < 20 ? "Below 20" :
                          n <= 50 ? "20 - 50" :
                          "Above 50"
              group n by range into g
              select g;
```

Rasheed

Prepa

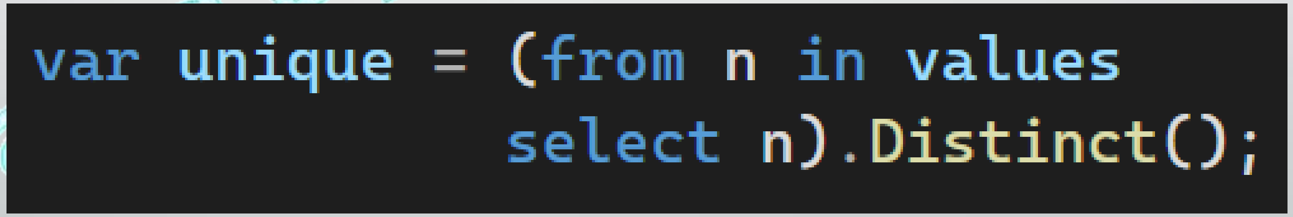
Distinct – Removing Duplicates

- Removes duplicate elements from a collection
- Method Syntax



```
var unique = values.Distinct();
```

- Query Syntax

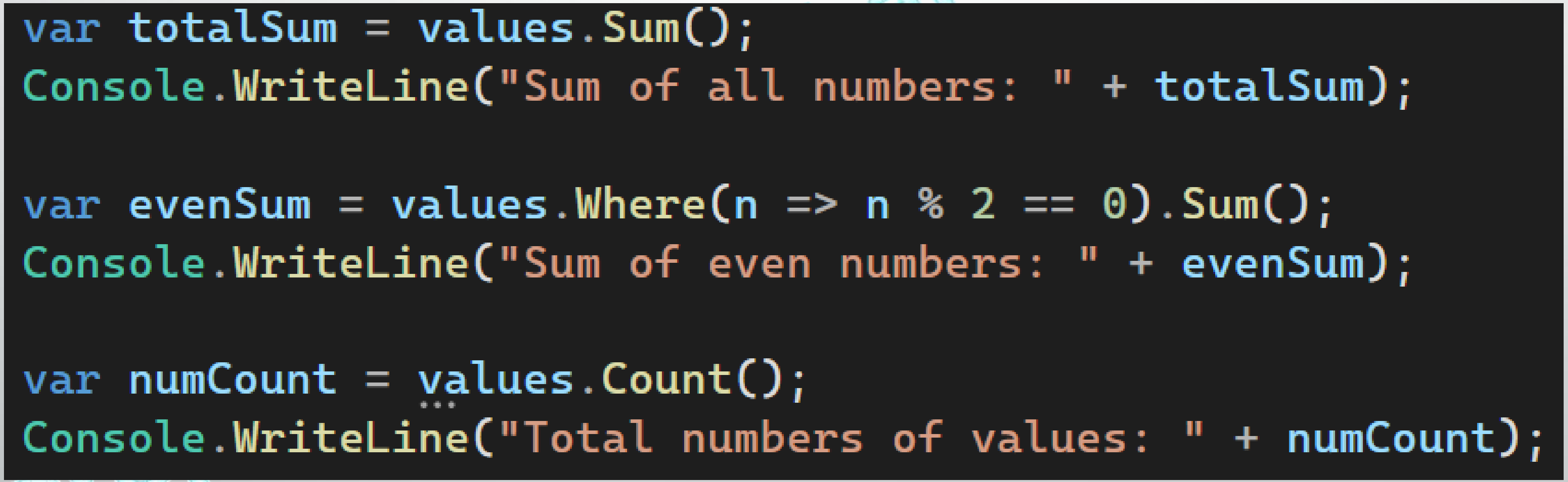


```
var unique = (from n in values
              select n).Distinct();
```

Prepared By: Ehtisham Rasheed

Sum, Count, Average, Min, Max – Aggregation

- Compute summary values
- Method Syntax



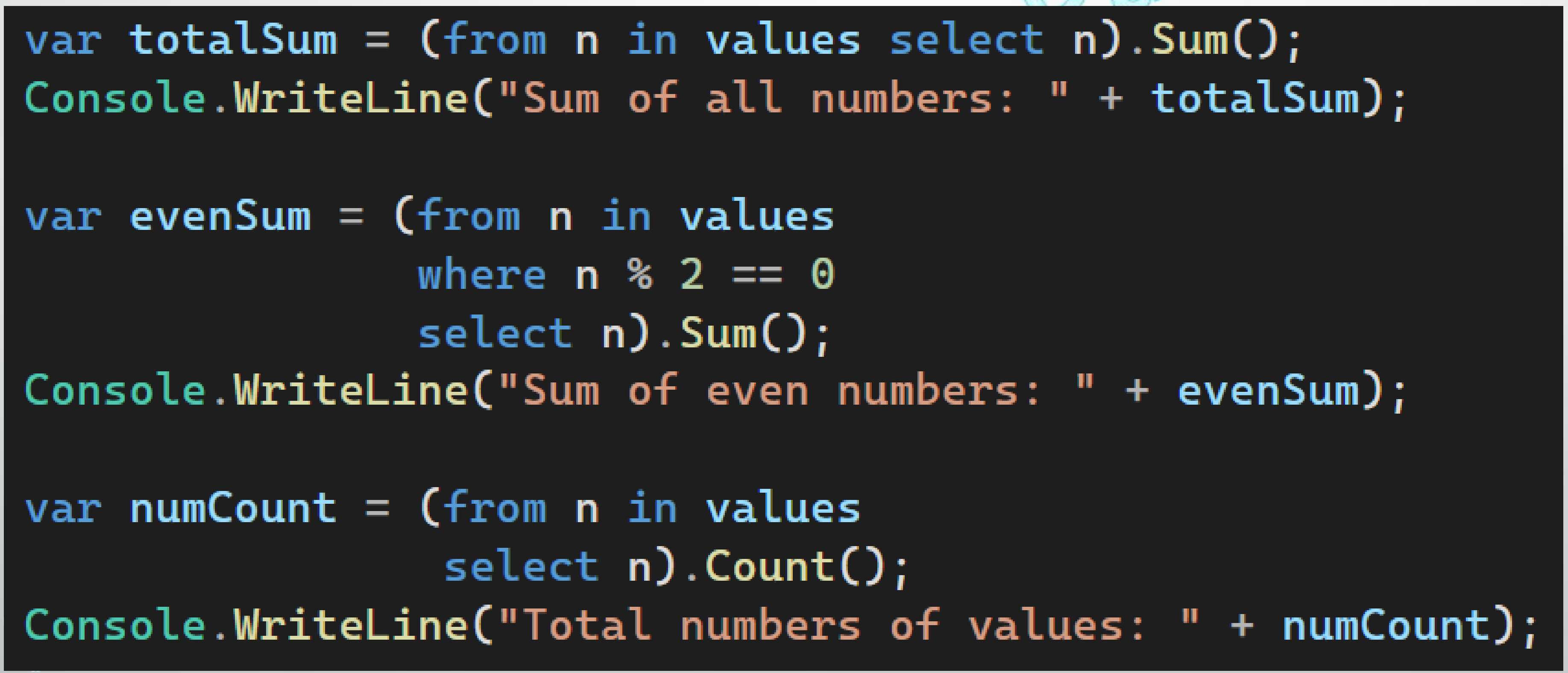
```csharp
var totalSum = values.Sum();
Console.WriteLine("Sum of all numbers: " + totalSum);

var evenSum = values.Where(n => n % 2 == 0).Sum();
Console.WriteLine("Sum of even numbers: " + evenSum);

var numCount = values.Count();
Console.WriteLine("Total numbers of values: " + numCount);
```

Pro... am Rasheed

Sum, Count, Average, Min, Max – Aggregation

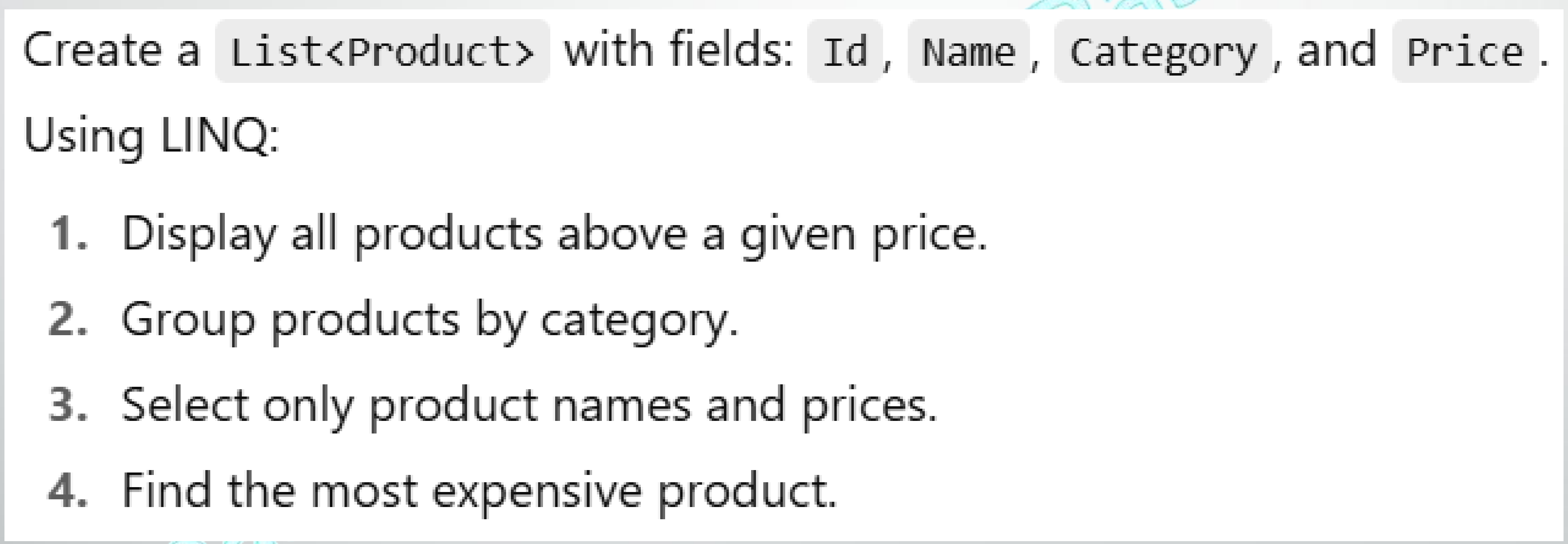
Rasheed

- Query Syntax



```
var totalSum = (from n in values select n).Sum();
Console.WriteLine("Sum of all numbers: " + totalSum);

var evenSum = (from n in values
               where n % 2 == 0
               select n).Sum();
Console.WriteLine("Sum of even numbers: " + evenSum);

var numCount = (from n in values
                select n).Count();
Console.WriteLine("Total numbers of values: " + numCount);
```

Exercise



Create a `List<Product>` with fields: `Id`, `Name`, `Category`, and `Price`.

Using LINQ:

1. Display all products above a given price.
2. Group products by category.
3. Select only product names and prices.
4. Find the most expensive product.

Prepa asheed