**LINQ – Practical Questions**

**LINQ with Method Syntax**

**Filter and Sort a List of Integers**

Create a list of random integers. Use LINQ to find all numbers greater than 50, sort them in descending order, and print the results.

**Find Specific Elements in a String List**

Given a list of names, use LINQ to find names that start with the letter "A" and print them.

Find the first name that has more than five characters.

**Project a List to New Values**

Create a list of numbers. Use LINQ to create a new list where each element is squared (e.g., 2 becomes 4). Print the original and transformed lists.

**Sum and Average Calculations**

Create a list of prices. Use LINQ to calculate and display the total sum and average price.

**Distinct Elements in a List**

Create a list of numbers with duplicates. Use LINQ to get only the unique elements and print the list.

**LINQ with Query Syntax**

**Filter and Project Properties**

Create a list of students with properties Name and Age. Use LINQ query syntax to select only students older than 18, then project their names into a new list and display it.

**Group by a Property**

Given a list of people with properties Name and City, use LINQ query syntax to group them by City and print each group along with the city name.

**Order by Multiple Properties**

Create a list of books with Title, Author, and Year properties. Use LINQ query syntax to order the books first by Author, then by Year, and display the sorted list.

**Select Elements with Conditions**

Create a list of employees with properties Name and Salary. Use LINQ query syntax to find employees with a salary above a certain threshold, and display their names and salaries.

**Elementary Operators with LINQ**

 **First and FirstOrDefault Operators**

* Create a list of ages. Use First to get the first age above 30, and FirstOrDefault to attempt getting an age above 100 (handle the result if not found).

 **Single and SingleOrDefault Operators**

* Create a list of product IDs. Use Single to retrieve a specific product ID, ensuring there’s exactly one match.
* Use SingleOrDefault to search for a non-existent product ID, and print an appropriate message if it’s not found.