LINQ EXAMPLES

Example no 1

List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

// Simple LINQ query to select numbers greater than 5

var result = from num in numbers

where num > 5

select num;

// Display result

foreach (var num in result)

{

Console.WriteLine(num);

}

Example no 2

List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

// Select numbers greater than 5

var result = numbers.Where(n => n > 5);

foreach (var num in result)

{

Console.WriteLine(num);

}

Example no 3

List<string> names = new List<string> { "John", "Jane", "Doe" };

// Select the lengths of each name

var nameLengths = names.Select(name => name.Length);

foreach (var length in nameLengths)

{

Console.WriteLine(length);

}

Example no 4

List<int> numbers = new List<int> { 7, 1, 3, 9, 2, 4 };

// Sort numbers in ascending order

var sortedNumbers = numbers.OrderBy(n => n);

foreach (var num in sortedNumbers)

{

Console.WriteLine(num);

}

Example no 5

List<int> numbers = new List<int> { 1, 2, 3, 4, 5 };

// Calculate the sum of all numbers

int sum = numbers.Sum();

Console.WriteLine("Sum: " + sum);

Example no 6

List<string> names = new List<string> { "Alice", "Bob", "Charlie" };

// LINQ query to find names that start with 'A'

var result = names.Where(name => name.StartsWith("A"));

foreach (var name in result)

{

Console.WriteLine(name);

}

Example no 7

Dictionary<int, string> students = new Dictionary<int, string>

{

{ 1, "John" },

{ 2, "Jane" },

{ 3, "Doe" }

};

// LINQ query to find students with keys greater than 1

var result = students.Where(s => s.Key > 1);

foreach (var student in result)

{

Console.WriteLine($"{student.Key}: {student.Value}");

}

Example no 8

List<int> numbers = new List<int> { 1, 2, 3, 4, 5 };

var evenNumbers = numbers.Where(n => n % 2 == 0);

foreach (var num in evenNumbers)

{

Console.WriteLine(num); // Outputs 2 and 4

}