LINQ Concepts with Simple Examples and foreach Loops

1. OfType<T>() - Filter by Type

```
ArrayList list = new ArrayList() { 1, "Hello", 2, "World" };
var intValues = list.OfType<int>();
foreach (var item in intValues)
{
    Console.WriteLine(item); // Output: 1, 2
}
```

2. OrderBy() + ThenBy() - Sorting

```
var products = new List<Product>
{
    new Product { Name = "Pen", Category = "Stationery", Price = 20 },
    new Product { Name = "Pencil", Category = "Stationery", Price = 10 },
    new Product { Name = "Apple", Category = "Fruit", Price = 50 },
    new Product { Name = "Banana", Category = "Fruit", Price = 30 }
};

var sorted = products.OrderBy(p => p.Category).ThenBy(p => p.Price);
foreach (var item in sorted)
{
    Console.WriteLine($"{item.Category} - {item.Name} - {item.Price}");
}
```

3. GroupBy() - Group Similar Items

```
var groupByCategory = products.GroupBy(p => p.Category);
foreach (var group in groupByCategory)
{
    Console.WriteLine($"Category: {group.Key}");
    foreach (var item in group)
    {
        Console.WriteLine($" {item.Name} - {item.Price}");
    }
}
```

4. Element Operators - Get Specific Items

```
var numbers = new List<int> { 10, 20, 30 };
Console.WriteLine($"First: {numbers.First()}");
Console.WriteLine($"FirstOrDefault: {numbers.FirstOrDefault()}");
Console.WriteLine($"Last: {numbers.Last()}");
Console.WriteLine($"ElementAt(1): {numbers.ElementAt(1)}");

foreach (var num in numbers)
{
    Console.WriteLine(num);
}
```

5. Set Operations - Compare Lists

```
--------------------------------------------------
var list1 = new List<int> { 1, 2, 3 };
var list2 = new List<int> { 3, 4, 5 };

var union = list1.Union(list2);
var intersect = list1.Intersect(list2);
var except = list1.Except(list2);

Console.WriteLine("Union:");
foreach (var item in union) Console.WriteLine(item);
Console.WriteLine("Intersect:");
foreach (var item in intersect) Console.WriteLine(item);
Console.WriteLine("Except:");
foreach (var item in except) Console.WriteLine(item);
```

6. Quantifier Operators - Check Conditions

```
--------------------------------------------------
var list = new List<int> { 2, 4, 6 };
Console.WriteLine($"All even? {list.All(n => n % 2 == 0)}");
Console.WriteLine($"Any > 5? {list.Any(n => n > 5)}");
Console.WriteLine($"Contains 4? {list.Contains(4)}");
foreach (var item in list)
{
    Console.WriteLine(item);
}
```

7. Aggregate Functions - Summary Values

```
--------------------------------------------------
var nums = new List<int> { 5, 10, 15 };
Console.WriteLine($"Count: {nums.Count()}");
Console.WriteLine($"Sum: {nums.Sum()}");
Console.WriteLine($"Min: {nums.Min()}");
Console.WriteLine($"Max: {nums.Max()}");
Console.WriteLine($"Average: {nums.Average()}");
foreach (var item in nums)
{
    Console.WriteLine(item);
}
```