



# SDEV2301

Lab Activity 3

LINQ Query Service + xUnit (TDD-lite)

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

# CONNECT (5 MINUTES)

# Quick Check

- Who remembers AAA?
- Who has used `Assert.Throws`?
- Who has compared two lists in a test before?

# EXPLORE (30 MINUTES)

# Explore 1 — What Is a Service Class?

## Domain Model

```
public class Product
{
    public string Name { get; set; } = "";
    public decimal Price { get; set; }
    public bool IsDiscontinued { get; set; }
}
```

## Before (what we did in Program.cs)

```
var active = products
    .Where(p => !p.IsDiscontinued)
    .ToList();
```

# After — Move Logic Into a Service

```
public class ProductQueryService
{
    public IEnumerable<Product> GetActiveProducts(IEnumerable<Product> products)
    {
        ArgumentNullException.ThrowIfNull(products);
        return products
            .Where(p => !p.IsDiscontinued)
            .ToList();
    }
}
```

# Explore 2 — Unit Testing Reset

## Value Assertions

```
Assert.Equal(expected, actual);  
Assert.True(condition);  
Assert.False(condition);  
Assert.NotNull(obj);
```



# Explore 2 — Unit Testing Reset

## Collection Assertions

```
Assert.Empty(list);
```

```
Assert.NotEmpty(list);
```

```
Assert.Equal(expectedCount, list.Count);
```

```
Assert.All(list, item => Assert.True(/* condition */));
```



# Explore 2 — Unit Testing Reset

## Exception Assertion

```
Assert.Throws<ArgumentNullException>(
    () => service.GetActiveProducts(null!)
);
```

[Fact]

```
public void GetActiveProducts_WhenProductsMixed_ReturnsOnlyActive()
{
    // Arrange
    var products = new List<Product>
    {
        new Product { Name = "Apples", Price = 3.5m, IsDiscontinued = false },
        new Product { Name = "Bananas", Price = 2.0m, IsDiscontinued = true }
    };
    var service = new ProductQueryService();
    // Act
    var result = service.GetActiveProducts(products).ToList();
    // Assert
    Assert.Single(result);
    Assert.False(result[0].IsDiscontinued);
}
```

# Explore 3 — How Do We Compare Lists?

## Pattern 1 -- Compare Key Sequences (Recommended)

```
var names = result.Select(p => p.Name).ToList();  
Assert.Equal(  
    new[] { "Apples", "Bananas", "Oranges" },  
    names  
);
```

# Explore 3 — How Do We Compare Lists?

## Pattern 2 -- Compare Projections

```
var pairs = result
    .Select(p => (p.Name, p.Price))
    .ToList();
Assert.Equal(
    new[]
    {
        ("Apples", 3.5m),
        ("Bananas", 2.0m)
    },
    pairs
);
```

# Explore 4 — What Are Guard Rails?

## Definition

- Guard rails = **input validation at the top of a method**
- Fail fast with a clear error.

## Example:

```
public IEnumerable<Product> SortByPrice(IEnumerable<Product> products)
{
    ArgumentNullException.ThrowIfNull(products);

    return products
        .OrderBy(p => p.Price)
        .ThenBy(p => p.Name)
        .ToList();
}
```

## Matching Test

```
Assert.Throws<ArgumentNullException>(
    () => service.SortByPrice(null!)
);
```

# Explore 5 — What Is a Tuple?

## Why Use Tuples?

- Sometimes we only need a few properties.
- Instead of creating a new class:

```
public IEnumerable<(string Name, decimal Price)>  
ProjectForListView(  
    IEnumerable<Product> products)  
{  
    ArgumentNullException.ThrowIfNull(products);  
    return products  
        .Select(p => (p.Name, p.Price))  
        .ToList();  
}
```



# How To Test Tuple Results

```
var result = service
    .ProjectForListView(products)
    .ToList();
Assert.Equal(
    new[]
    {
        ("Apple", 3.5m),
        ("Banans", 2.0m)
    },
    result
);
```

# PRACTICE (60 MINUTES)

# Suggested Pacing

- Minute 0-10 --> Write one test (US1)
- Minute 10-20 --> Implement service method
- Minute 20-35 --> Sorting test + deterministic order
- Minute 35-50 --> Tuple projection
- Final --> Guard rails

# ASSESS

**WE  
ARE** **ESSENTIAL  
TO ALBERTA**



# INSTRUCTOR CHECKLIST

- At least 4 passing tests (core stories)
- Evidence of at least one RED --> GREEN cycle
- No mutation of input collections
- Deterministic sorting (tie-breaker included)

Use 0–5 rubric.

# REFLECT

**WE  
ARE** ESSENTIAL  
TO ALBERTA



# Closing

Today you:

- Used a service class
- Wrote behavior-based unit tests
- Compared list results
- Used guard rails
- Returned tuples