

# Language Features Part 2

K. Scott Allen  
odetocode.com  
@OdeToCode



**pluralsight**   
hardcore dev and IT training

# Declaration Expressions

```
int result = 0;
var odd = numbers.Where(n => n % 2 == 1).ToList();
foreach (var n in odd)
{
    result += n + odd.Count();
}
return result;
```

```
int result = 0;
foreach (var n in var odd = numbers.Where(n => n % 2 == 1).ToList())
{
    result += n + odd.Count();
}
return result;
```

# Using Static

```
class Program
{
    public static void Main()
    {
        Console.WriteLine("Hello World!");
    }
}
```

```
using System.Console;

class Program
{
    public static void Main()
    {
        WriteLine("Hello World!");
    }
}
```

# Errors, Nulls, Exceptions

```
var name = operation ?.Method ?.Name ?? "no name";

try
{
    operation();
    await _logWriter.WriteAsync(name + " executed");
}
catch (Exception ex) if (ex.Message != null)
{
    await _logWriter.WriteAsync(name + " failed");
}
finally
{
    await _logWriter.FlushAsync();
}
```

# nameof

```
public void DoWork(string name)
{
    if(name == null)
    {
        throw new ArgumentNullException("name");
    }
}
```

```
public void DoWork(string name)
{
    if(name == null)
    {
        throw new ArgumentNullException(nameof(name));
    }
}
```

# Expression Bodied Members

```
public class Point(int x, int y)
{
    public int X => x;
    public int Y => y;
    public double Distance => Math.Sqrt(x*x + y*y);
    public Point Move(int dx, int dy) => new Point(x + dx, y + dy);
}
```

# Summary

```
using System.Console;

class Program
{
    public static void Main()
    {
        WriteLine("Hello World!");
    }
}
```

```
var name = operation ?.Method ?.Name ?? "no name";

try
{
    operation();
    await _logWriter.WriteAsync(name + " executed");
}
catch (Exception ex) if (ex.Message != null)
{
    await _logWriter.WriteAsync(name + " failed");
}
finally
{
    await _logWriter.FlushAsync();
}
```