MVA jumpstart

Programming in C# Jump Start

Jerry Nixon | Microsoft Developer Evangelist
Daren May | President & Co-founder, Crank211

Microsoft

Meet Jerry Nixon | Colorado

- Microsoft Developer Evangelist
  - Reaching Professional & Student Communities
  - Teaching Developers about Windows Phone
  - Teaching Developers about Windows 8
  - Teaching Developers about XAML

http://jerrynixon.com
@jerrynixon

Meet Daren May | @darenmay

- President & Co-founder, Crank211
  - Specializes in designing and building next-level digital experiences
- Prior to starting Crank211…
  - CTO and VP Operations for software consultancy
  - Practice lead for EMC Consulting
  - Leveraged Microsoft technologies since Visual Studio .NET betas
  - Developing XAML-based solutions since the heady days of "Avalon" (the early name for WPF)

Agenda for the Day

First Half | Features of C#	Second Half | Advanced C#
01 | Object Oriented Programming, Managed Languages and C#	05 | Advanced C#, Type and Value Validation; Encryption Techniques
02 | Constructing Complex Types; Object Interfaces and Inheritance	06 | Splitting Assemblies and WinMD; Diagnostics and Instrumentation
03 | Code Reflection and Information; Working with Garbage Collection	07 | Interacting with the File System; Leveraging Web Services
04 | Controlling Programmatic Flow; Manipulating Types and Strings	08 | Using LINQ to Objects and XML; Fundamentals of Serialization
** MEAL BREAK **	

Setting Expectations

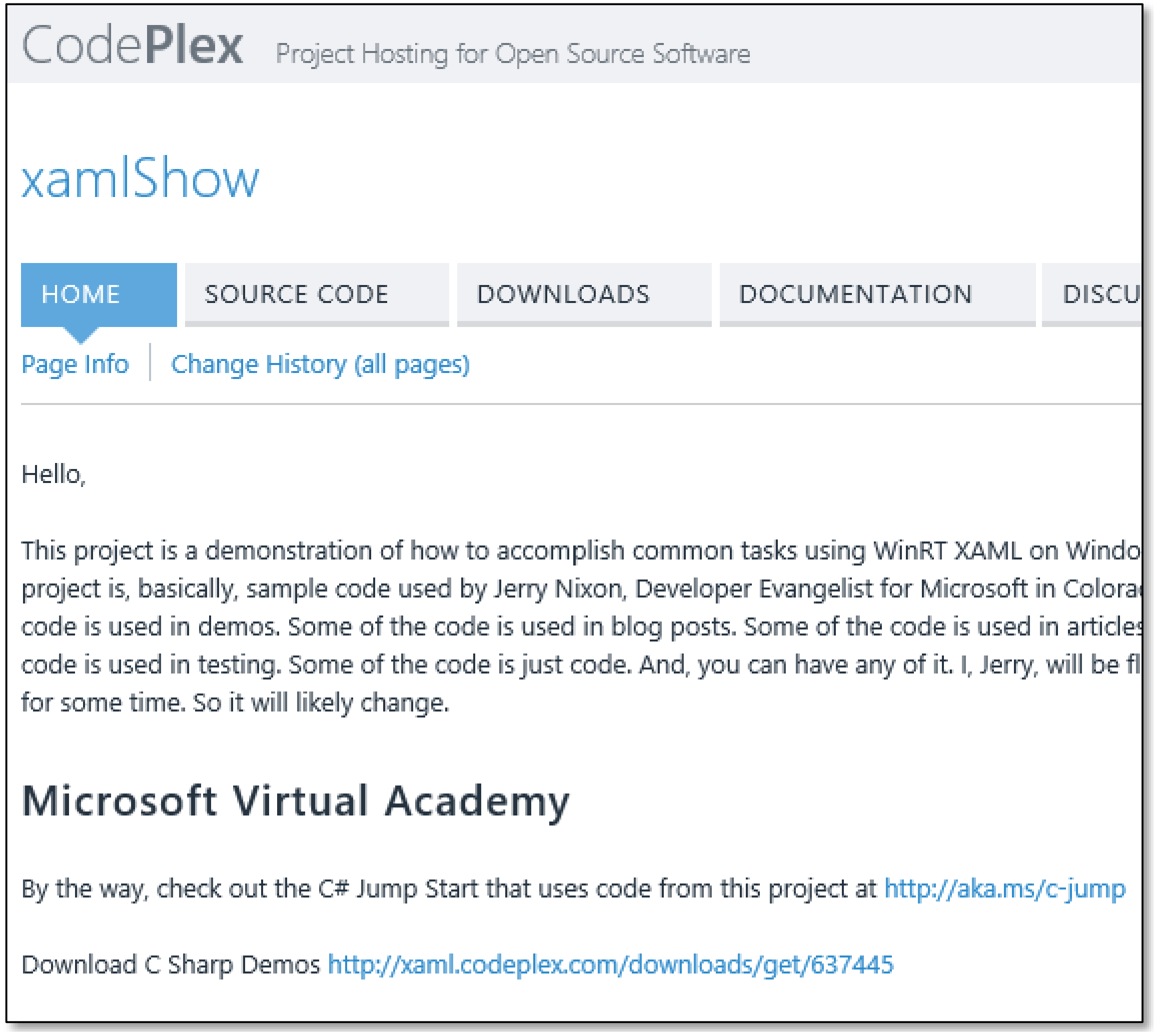
- Seasoned to Intermediate Developers
  - Accelerated coverage of C# fundamentals expected on Microsoft Exam 70-483
- Suggested Prerequisites/Supporting Material
  - Accelerated version of Microsoft course 20483B, a five-day course (which includes hands-on labs).
  - We recommend developers new to C# or who need hands-on experiences take this course with a Microsoft Learning Partner

Join the MVA Community!

- Microsoft Virtual Academy
  - Free online learning tailored for IT Pros and Developers
  - Over 1M registered users
  - Up-to-date, relevant training on variety of Microsoft products
- "Earn while you learn!"
  - Get 50 MVA Points for this event!
  - Visit http://aka.ms/MVA-Voucher
  - Enter this code: **ProgC#Jump** (expires 3/29/2013)

Download Code Samples

- Visit xamlShow on CodePlex
  - http://xaml.codeplex.com
  - Bookmark this link
- "Microsoft Virtual Academy" section…
  - Click "Download C Sharp Demos" link
  - This link will change, so click the link from http://xaml.codeplex.com each time


CodePlex Project Hosting for Open Source Software
xamlShow
HOME
SOURCE CODE
DOWNLOADS
DOCUMENTATION
DISCU
Page Info
Change History (all pages)

Hello,

This project is a demonstration of how to accomplish common tasks using WinRT XAML on Windo project is, basically, sample code used by Jerry Nixon, Developer Evangelist for Microsoft in Colora code is used in demos. Some of the code is used in blog posts. Some of the code is used in articles code is used in testing. Some of the code is just code. And, you can have any of it. I, Jerry, will be fl for some time. So it will likely change.

Microsoft Virtual Academy

By the way, check out the C# Jump Start that uses code from this project at http://aka.ms/c-jump

Download C Sharp Demos http://xaml.codeplex.com/downloads/get/637445

MVA jumpstart

01 | Object Oriented Programming, Managed Languages and C#

Jerry Nixon | Microsoft Developer Evangelist
Daren May | President & Co-founder, Crank211

Module Agenda

- Object Oriented Programming
- What is a Managed Languages
- Why C# for OOP?

Object Oriented Programming

MVA Microsoft Virtual Academy

Microsoft

What is an Object?

- An object typically models a concept:
  - An object usually “is” something – i.e. a customer
  - An object usually “has” data – i.e. the customer’s first name
  - An object usually “performs” actions – i.e. make a customer “preferred”

What is Object Oriented Programming?

- To be object oriented, a language is designed around the concept of objects. that are something, have certain properties and exhibit certain behaviors. This means the language generally includes support for:
  - Encapsulation
  - Inheritance
  - Polymorphism

What is a Managed Language?

MVA Microsoft Virtual Academy

Microsoft

What is a Managed Language?

- Managed languages depend on services provided by a runtime environment.
- C# is one of many languages that compile into managed code.
- Managed code is executed by the Common Language Runtime (CLR).
- The CLR provides features such as:
  - Automatic memory management
  - Exception Handling
  - Standard Types
  - Security

Why Standard Types

- Foundational building block is a *Type*
  - Metadata about space allocation
  - Metadata for compile-time type checking
- All types have a common base - *Object*
  - Object defines ToString, so every object has a ToString function
- There are 3 categories of types:
  - Value types – these directly store values
  - Reference types – or objects, store a reference to data
  - Pointer types – only available in unsafe code

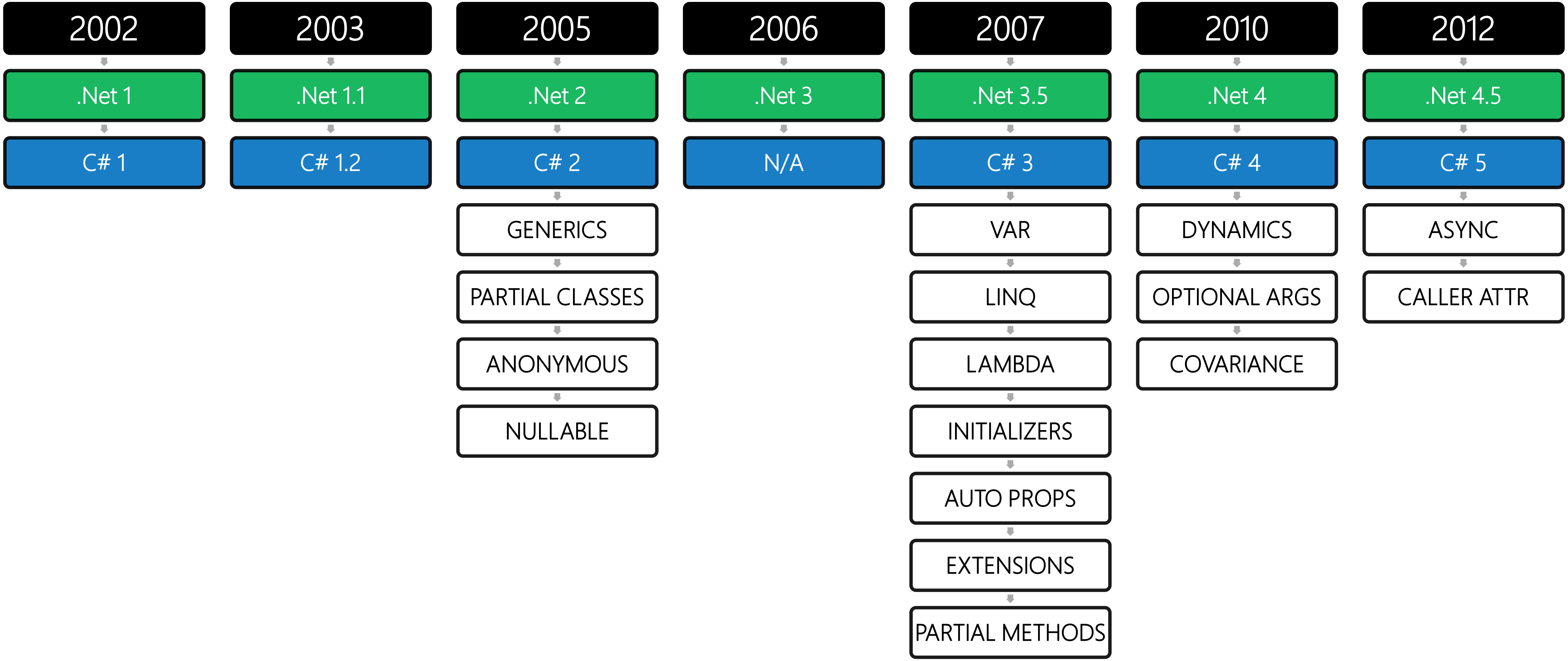
Why C# for OOP?

MVA Microsoft Virtual Academy

Microsoft

Why Use C#?

- C# (pronounced "C sharp") is a programming language that is designed for building a variety of applications.
  - General Purpose
  - Object Oriented
  - Focused on dev productivity
- C# has evolved over time to include many new features:



2002	2003	2005	2006	2007	2010	2012
.Net 1	.Net 1.1	.Net 2	.Net 3	.Net 3.5	.Net 4	.Net 4.5
C# 1	C# 1.2	C# 2	N/A	C# 3	C# 4	C# 5
		GENERICS		VAR	DYNAMICS	ASYNC
		PARTIAL CLASSES		LINQ	OPTIONAL ARGS	CALLER ATTR
		ANONYMOUS		LAMBDA	COVARIANCE	
		NULLABLE		INITIALIZERS		
				AUTO PROPS		
				EXTENSIONS		
				PARTIAL METHODS		

C# & Encapsulation

- Encapsulation means to *create a boundary* around an object to separate its external (public) behavior from its internal (private) implementation.
- Consumers of an object should only concern themselves with what an object does, *not how it does it*.
- C# supports encapsulation via:
  - Unified Type System
  - Classes and Interfaces
  - Properties, Methods and Events

C# & Inheritance

- C# implements Inheritance in two ways:
  - A class may inherit from a single base class
  - A class may implement zero or more Interfaces

C# & Polymorphism

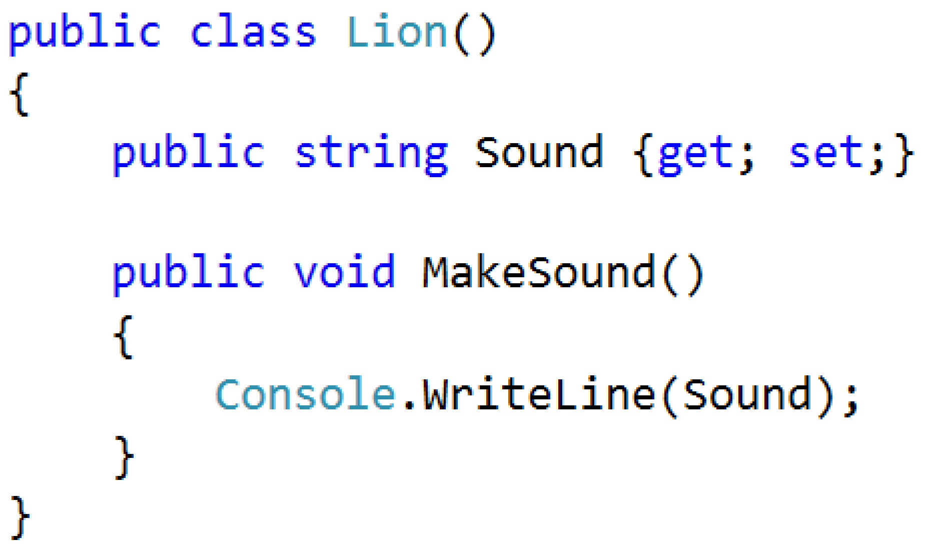
- A class can be used as its own type, cast to any base types or interface types it implements.
- Objects may methods as virtual; virtual methods may be overridden in a derived type. These are executed instead of the base implementation.

Developer Productivity

- From the outset, C# focused on making it easier for developers to solve complex tasks without compromising elegance.
- Examples :
  - var - simplifies variable definition while retaining strong typing
  - LINQ – language integrated query
  - Lambdas – a further refinement of anonymous methods used extensively in LINQ

C# Syntax

- C# syntax is based on the C & C++ syntax.
- Identifiers are names of classes, methods, variables, and so on:
  - Lion, Sound, MakeSound(), Console, WriteLine()
- Keywords are compiler reserved words:
  - public, class, string, get, set, void



```
public class Lion()
{
    public string Sound {get; set;}

    public void MakeSound()
    {
        Console.WriteLine(Sound);
    }
}
```

What is Code Decoration?

- Attributes
  - – Associate additional metadata to types and members
  - – Discoverable at runtime via reflection
- Does C# still need comments?
  - – Block comments
  - – Single line comments
  - – XML documentation
    - Can be extracted into separate XML files during compilation
    - Can be used to generate formal documentation

What are Lambda Expressions?

- An enhancement of anonymous methods
- An unnamed method written inline
- An alternative to a delegate
- At compile time a lambda expression becomes either:
  - An instance of a delegate
  - Or an expression tree
- Expressions are validated at compile time, but evaluated at run time.
- Dynamic expression trees are supported

What are Anonymous Types?

- Types without a design-time definition
  - Used extensively with LINQ
- Compiler inferred types
- Is it portable?

Extension Methods

- Extension methods extend types without altering them.
  - Especially useful for extending classes that are not yours or are sealed
- They are declared as static methods in a static class.
- The first parameter has the *this* modifier
- The first parameter's type is the type being extended

What are Dynamics?

- Instructs the compiler to ignore type checking at compile-time
- Defers type checking until runtime
- Simplifies interaction with COM interfaces / un-typed, external objects
- Why not use it all the time?

DEMO

Create and compile simple console app

Module Recap

- Object Oriented Programming
- What is a Managed Languages
- Why C# for OOP?

Microsoft