OOP Concepts in CFML

Nolan Erck
Director / Owner
South of Shasta

About Me

- Owner @ South of Shasta
  - Developer Consultancy
- Organizer @ Sac Interactive
- Recovering Video Game Developer (Grim Fandango, Sim Park, Star Wars Phantom Menace, etc)
- Music Junkie


I'M REAL PROUD OF MY RECORD COLLECTION

Today’s Agenda

- Why OOP?
- Your First CFC
- Pseudo Constructors & Real Constructors
- Inheritance
- Composition
- Tips and Other Resources

First things first…

OOP is Hard

- New concepts.
- New way of thinking and organizing code.
- Terminology is different.
- Lifespan and scope of variables, design patterns, many new things.
- Not all if it makes total sense at first.

That is totally normal!

Second things first…

STOP over-using CFinclude!

(Incorrectly anyway.)

(Demo 1)

Why OOP?

- Industry standard for literal decades.
- Translates between languages.
- Better design for larger projects.
- More reusability.
- More self-contained code blocks.
- Easier to split up the work using “design patterns”.

OOP is an Industry Standard

- Not just CFML standard.
- All modern development.
- Can trade design and architecture ideas with non-CFML shops.
- Senior development is not “I know more syntax”. It’s knowing proper software architecture patterns and rules for safely writing code.
- “I’ve been writing code for 20 years, I am a senior developer”.
    - Maybe, maybe not.

CFC Myths

- CFCs means I have to rewrite my entire app.
- CFCs have security problems.
- CFCs are slow.
- I have to use a framework.

Your first CFC

(Demo 2)

Pseudo Constructors

(Demo 3)

Pseudo Constructors

- Bad practice.
- Too easy to lose track of where the code is in a CFC.
- No way to pass arguments in/out of that code block.
- Doesn’t follow the standard way objects are init’d in other languages.
- Use real Constructors instead.

Constructors

- OOP standard.
- How you initialize an object when it’s created.
- new() or CreateObject() or <cfobject>
  - All work, in slightly different ways
  - new() is industry standard
- init() is the constructor in CFML
  - (Each language has their own variation on this.)

Constructors

(Demo 4)

Objects & Instances

(Demo 5)

Public vs Private

(Demo 6)

Inheritance

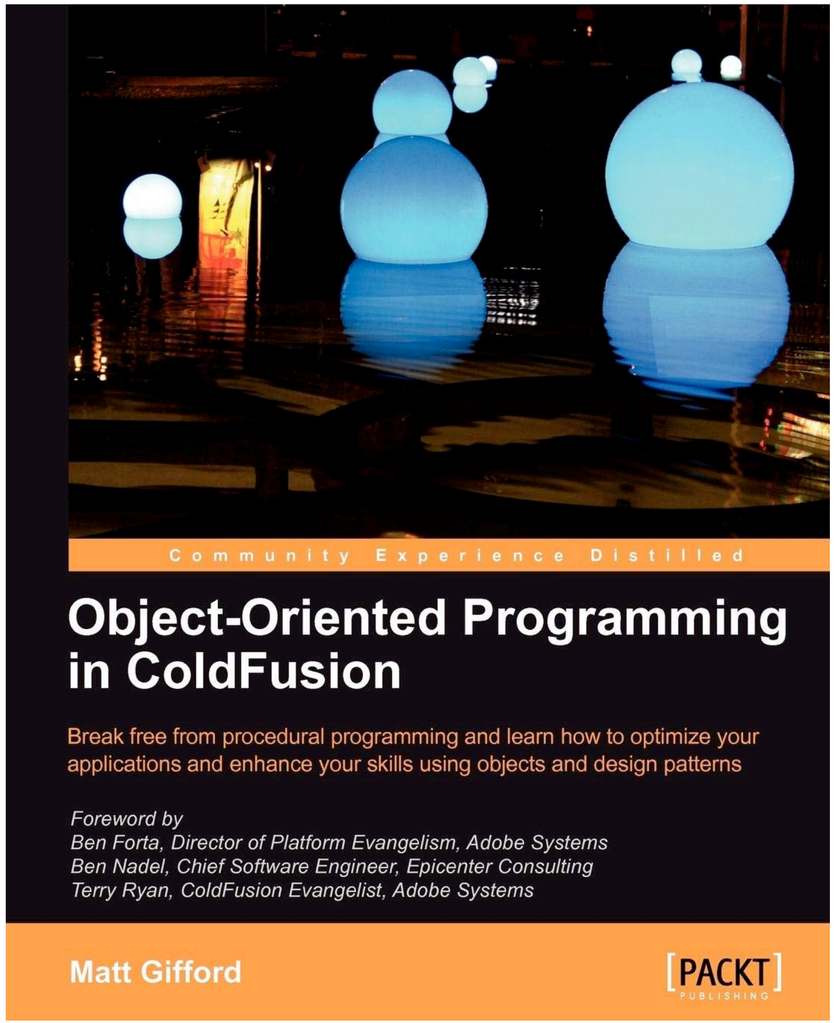
(Demo 7)

Composition

(Demo 8)

Matt Gifford's Book

- Object Oriented Programming in ColdFusion
- Solid OOP Primer
- Cover's some basic uses for CFCs
- Also easy to follow patterns like DAO, Gateway, Beans, etc.
- Don't let the publishing date fool you, the content (especially the concepts) is all still valid.

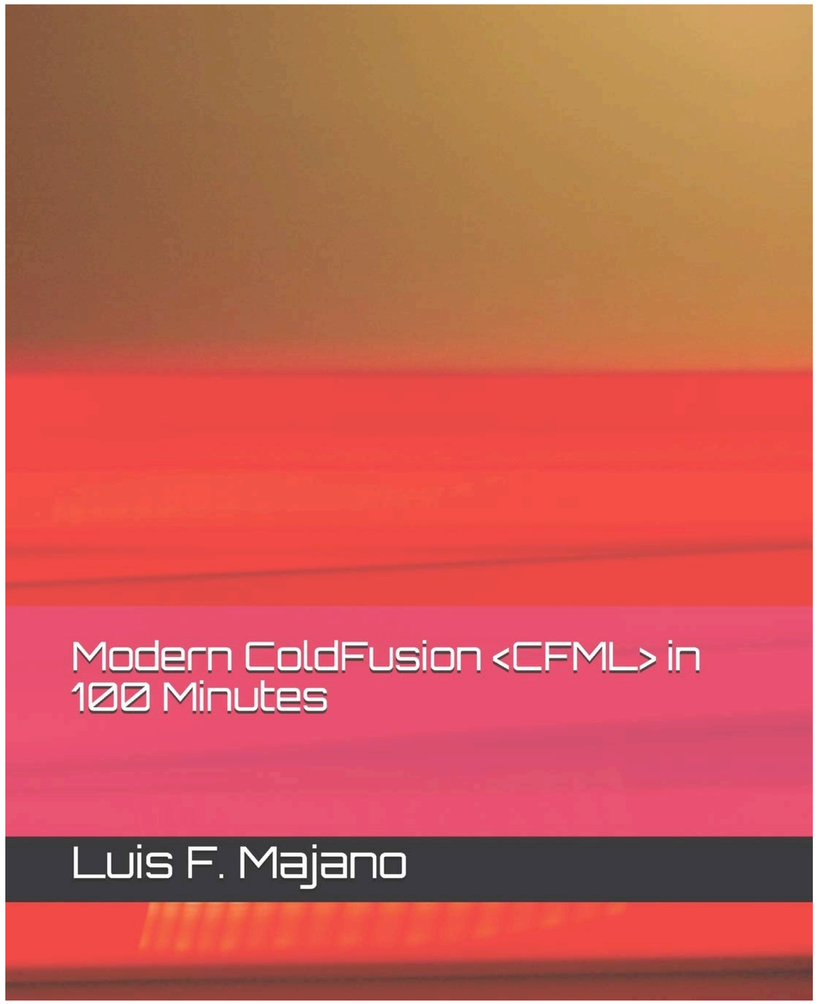


The Head First Book

- Head First Design Patterns
- The BEST book on design patterns ever!
- Code samples in Java, but that doesn't matter.
- I use this any time I teach an OOP workshop, even if it's not a Java class.

Modern CFML In 100 Minutes

- By Luis Majano
- Covers a variety of tips for writing modern CFML.
- Some OOP, some just general best practices and useful info.
- Very quick read.



More Resources

- CFCasts.com
- The CF Summit Conference, next week!
- ColdFusion Certification training materials from Adobe.
- South of Shasta provides training.
- LearnCFInAWeek.com

Learning in 30 minutes a day…

Thanks!

- You can find me here:
  - southofshasta.com
  - Twitter (for now) @southofshasta, @nolanerck
  - “South of Shasta” channel on YouTube
  - nolan@southofshasta.com