

A Manager's Guide To  
**MOOSE**

(Heavily derived from Stevan Little's Talk of the same name.)

# Cory G Watson: 'gphat'

- Director, Magazines.com
- Creative Director, Cold Hard Code
- Catalyst, DBIx::Class, Graphics::Primitive
- <http://www.onemogin.com/blog/>
- JAPH

# WHAT IS MOOSE?

# WHAT MOOSE IS NOT.

# What Moose Is Not

- An experiment or prototype.
- A toy.
- Yet Another Accessor Builder
- A source filter.
- Perl 6 in Perl 5.

# What Moose Is

- An extension of Perl 00
- Stable and production ready.
- Lessons learned from lots of other languages.
- A step toward Perl 6.

# Class::MOP

- CLOS for Perl 5
- Formalization of Perl 5 OO.
- 20 year old Lisp technology.
- Moose is syntactic sugar.

# Simple Example

```
package Shape;  
use Moose;
```

```
has 'x' => (  
    is => 'rw',  
    isa => 'Int',  
    default => 0  
);
```

# What Did That Do?

- Created a class.
- Used 'strict' and 'warnings'.
- Added an 'x' attribute.
- Made 'x' read-write.
- Requires a type of 'Int'.
- Defaults 'x' to 0.

# How Is It Used?

```
my $shape = Shape->new(  
    x => 5  
);  
say $shape->x;  
$shape->x(6);
```

# More OO

```
package Rectangle;  
use Moose;
```

```
extends 'Shape';
```

```
has 'height' => ...
```

# Extends?

- Subclasses the given class.
- Lets you add new attributes.
- Allows you to modify existing attributes...

# Modify?

```
has '+x' => (
  default => 5
);
```

**NEAT!**

**HELL YES.**

# Type-ography

- Moose has lots of types.
- Types are subclassable.
- Any class works (`isa => 'MyClass'`)
- `ArrayRef[MyClass]` DWIMs
- `Maybe[MyClass]`, could be undef

# Meta-Shmeta

- All classes have a `->meta` (may change)
- `->meta` is a metaclass
- `Shape->meta->name # Shape`
- `Shape->meta->attribute_list # [ x ]`
- `Rectangle->meta->superclasses # [ Shape ]`

# Phenomenal Meta Power

- `meta->add_method(...);`
- `meta->remove_method(...);`
- `meta->add_attribute(...);`
- `meta->remove_attribute(...);`

**BUT...**  
**WHY?**

# Why?

- Perl is Old and Ugly
- OO involves **lots** of error-prone yak shaving.
- CPAN is full of junk.
- Line noise.

# TMTOWTDI

There's More Than One Way To Do It.

# BCINABT

But Consistency Is Not A Bad Thing.

90%

Percentage Of Ways That Are Wrong.\*

\* Completely made up stat. Provided for laughs.

**THAT'S A  
PROBLEM?**

**HELL YES.**

# Hard to Hire Perl.

- No keywords to scan for.
- Are they scripters?
- Programmers must interview.
- Must translate!

# OOP Madness

Class::Accessor Class::MakeMethods base.pm Spiffy  
Class::HPLOO Class::Base Object::Tiny Object::Lexical  
EO Class::Accessor::Fast Class::Closure Class::Meta  
Class::Simple Class::Gomor Rose::Object Class::Builder  
Class::InsideOut Object::LocalVars Oak::Object OOP  
Object::InsideOut Class::Dot Class::NamedParms Myco  
Class::Structured Class::Classless parent.pm Eobj  
Class::Prototyped Class::Init Class::Maker Class::Object  
Fukurama::Class Class::Declare Class::Std Object::Declare  
Class::Struct Class::AutoClass Class::Root Badger Oryx  
Object::Prototype Basset Object::Accessor Class::Lego  
Class::Container Tangram OO::Closures Class::Trait MOP  
Object::MultiType SLOOPS Class::TOM Class::PObject  
Moose

# WHY MOOSE?

# Pedigree

- CLOS
- Smalltalk
- Perl 6 meta model

# Consistency

- Basis for Hiring.
- Outside interaction (#moose)
- Breeds Maintainability.

# Extensible

- `MooseX::*`
- Lots of awesome stuff.
- A place for your developers to put their ideas.

# Free Shit

- MooseX::*\**
- Lots of awesome stuff.
- TESTING!!!!111oneeleven

# Incremental

- Not all-or-nothing.
- Start with single classes.
- Weave into other places.
- Use where it makes sense.

# QUESTIONS?