Object Orientated Programming

OOP - ActionScript 2 and 3
JavaScript, C++, C# and Objective C

Basic Concepts of OOP

Linear Programming

```
10 REM *******************
20 REM * Tax Calculation *
30 REM *******************
40 CLS
50 print “Welcome to TAXCalc 1970”
60 print
70 input “Please enter amount
     to calculate:”;amount
80 input “Please enter product
     code: “;product$
90 newTax = amount * 0.175
```

Basic Concepts of OOP

Linear Programming

Xerox PARC

Alan Kay

Alan Curtis Kay (born May 17, 1940) is an American computer scientist, known for his early pioneering work on object-oriented programming and windowing graphical user interface design. He is the president of the Viewpoints Research Institute, and an Adjunct Professor of Computer Science at the University of California, Los Angeles. He is also on the advisory board of TTI/Vanguard. Until mid 2005, he was a Senior Fellow at HP Labs, a Visiting Professor at Kyoto University, and an Adjunct Professor at the Massachusetts Institute of Technology.

Basic Concepts of OOP

Linear Programming

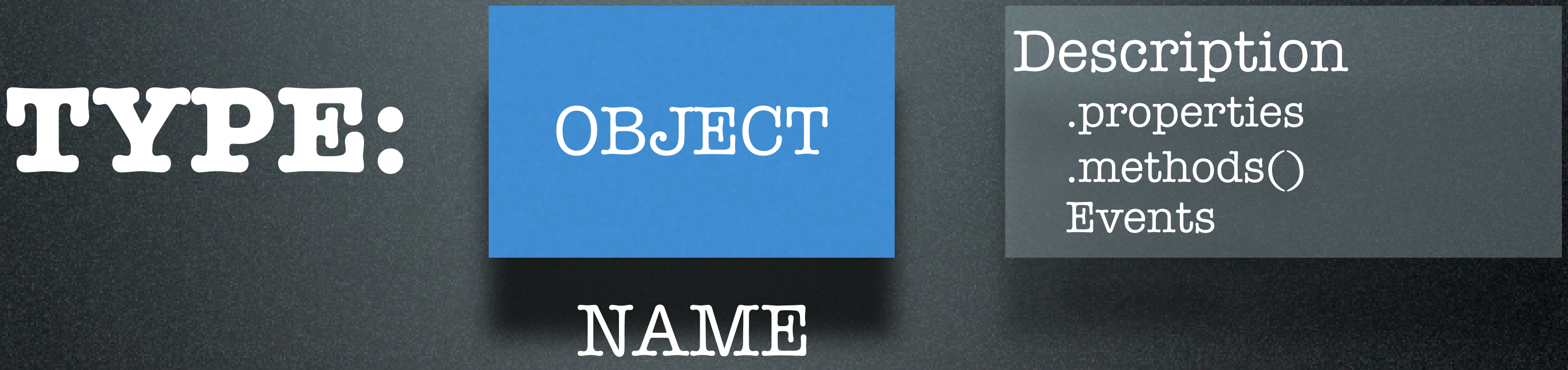
Basic Concepts of OOP

Linear Programming

Object Orientated Programming

Basic Concepts of OOP

Object Orientated Programming



TYPE:

OBJECT

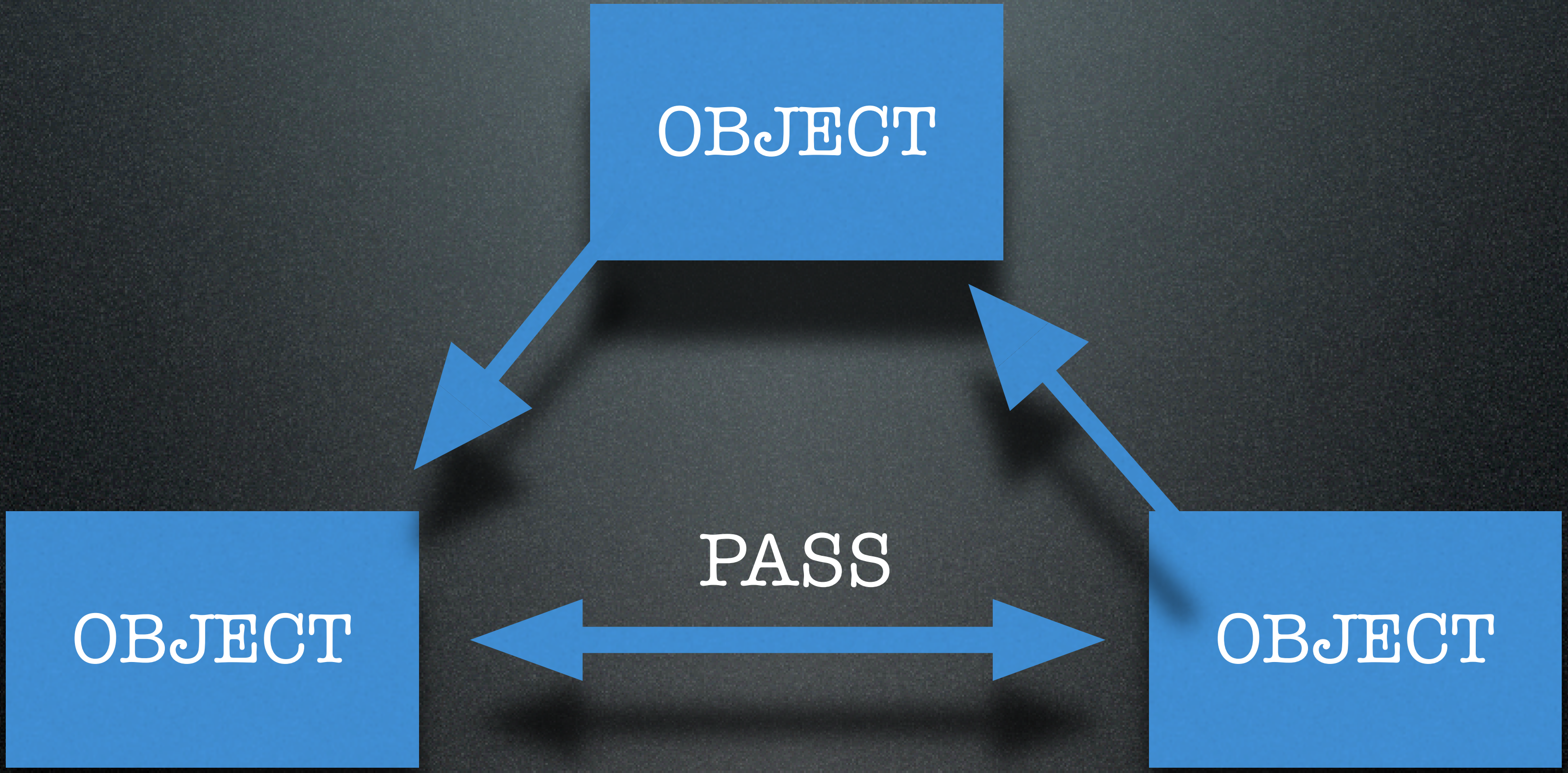
NAME

Description

- .properties
- .methods()
- Events

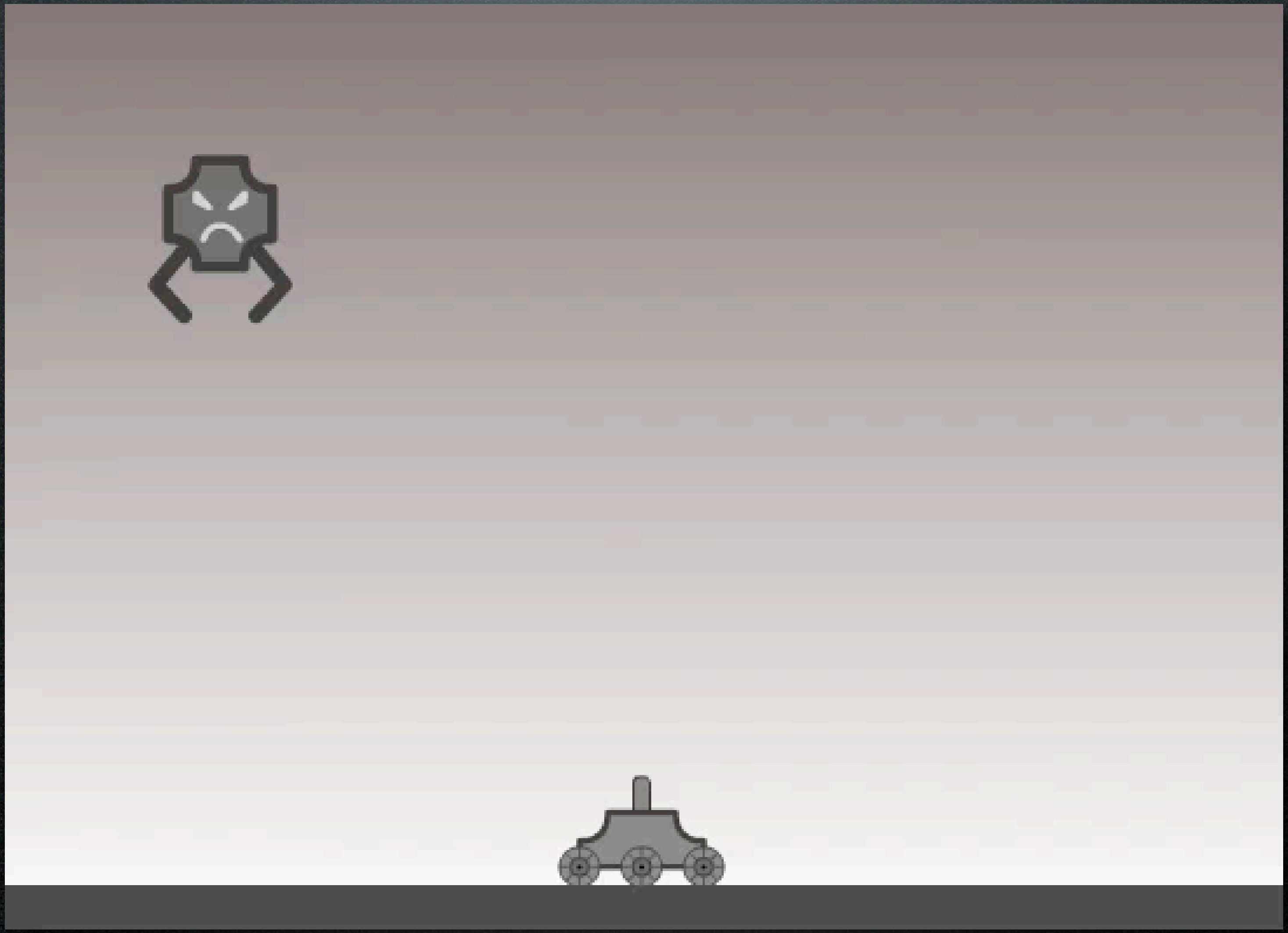
Basic Concepts of OOP

Object Orientated Programming



Basic Concepts of OOP

Object Orientated Programming



Basic Concepts of OOP

Object Orientated Programming

CLASS SYSTEM

CLASS

Classification

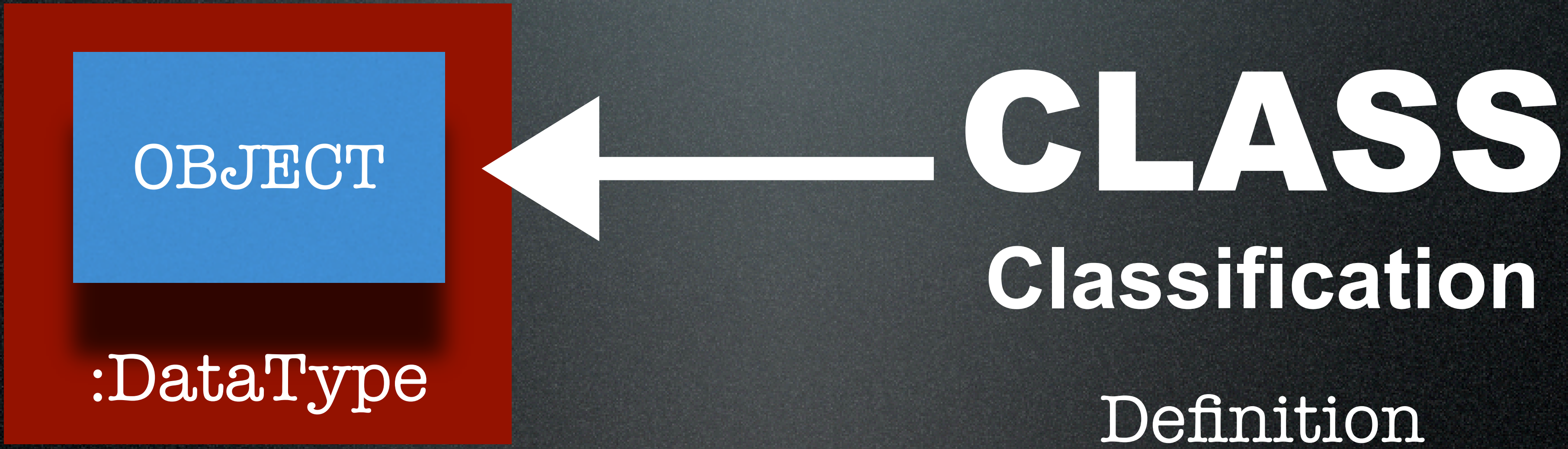
OBJECT

Basic Concepts of OOP

Object Orientated Programming

CLASS SYSTEM

Variable



OBJECT

:DataType

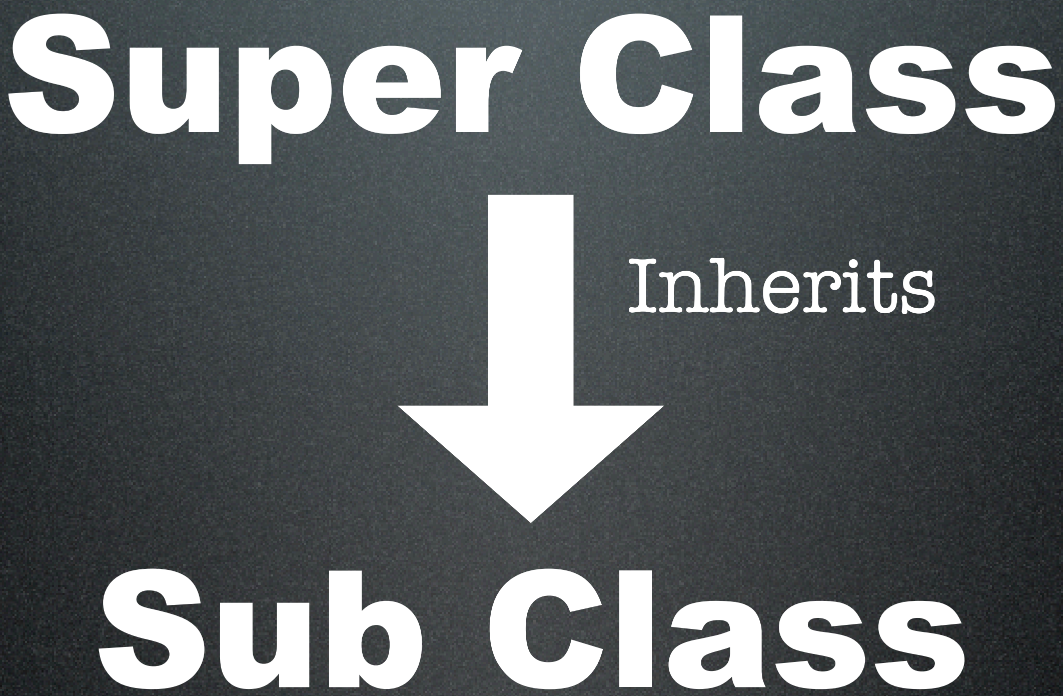
CLASS

Classification

Definition

Instance of a Class

Class System



Super Class

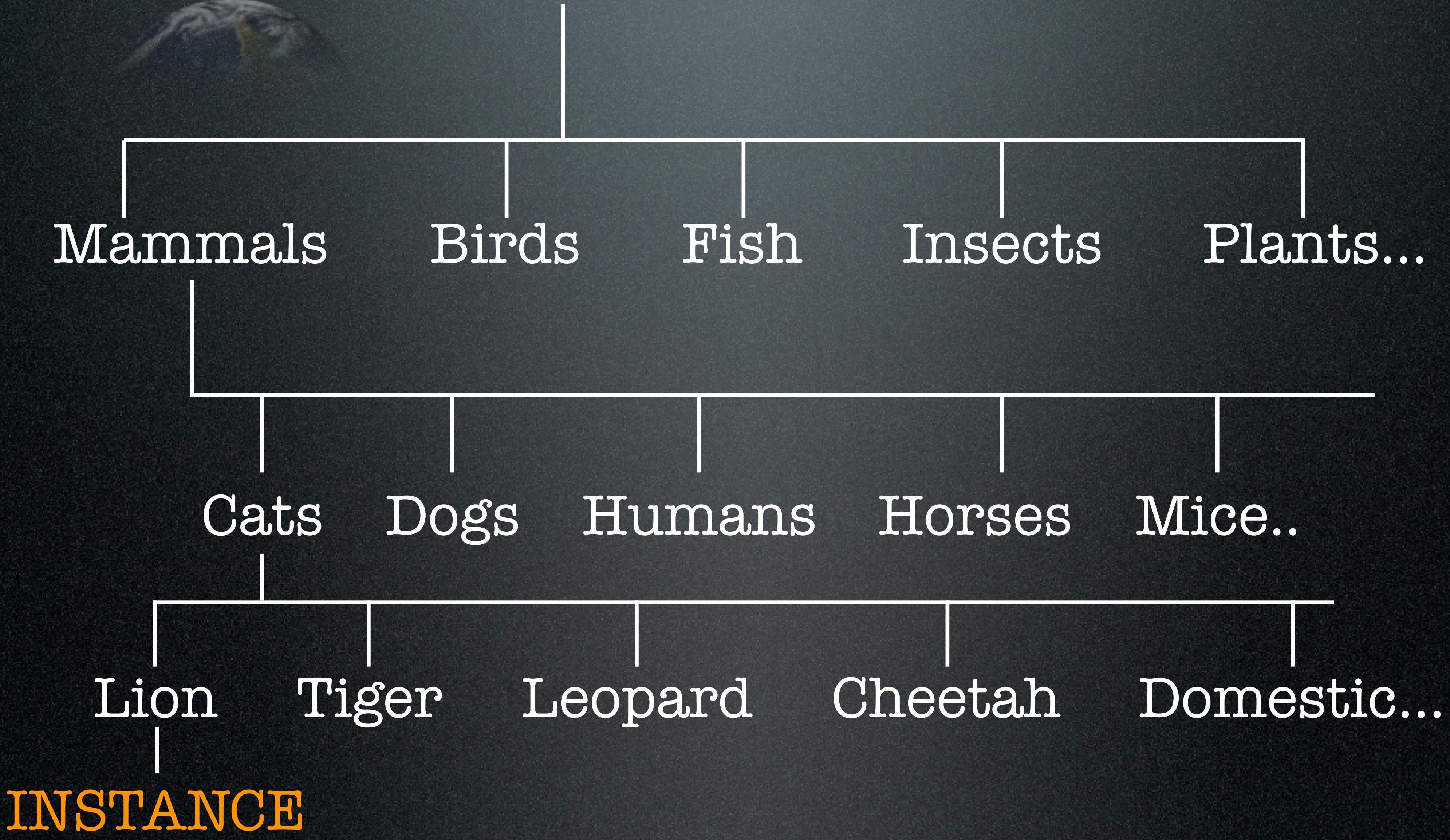
Inherits

Sub Class

Class System

Living

Inanimate



Mammals

Birds

Fish

Insects

Plants...

Cats

Dogs

Humans

Horses

Mice..

Lion

Tiger

Leopard

Cheetah

Domestic...

INSTANCE

Working with Classes

1. Look up what class we need to use

2. Create an instance of the class

3. Set up instance and control it

Working with Classes

all Classes start with a **C**apital letter

have no_spaces_in_their_name

usually written:

LikeThis

Instance Names

always start with a lowercase letter

NEVER! a number

Numbers can be used

noSpaces

myInstance

Working with Classes

Create an instance of a Class

1. use the var command to create a new variable

2. need to DataType

3. ask the Class to make a new instance and put it in the variable

Working with Classes

```
var myInstance:DataType = new Class();

var myTiger:Tiger = new Tiger();

var myTiger:Tiger = new Tiger(“Bengal”);

var myMovie:MovieClip = new MovieClip();

var myImage = new Image();  [JavaScript]
```

Working with Classes

CLASS

Properties

Methods

Events

Working with Classes

CLASS

Properties:

width	_width
height	_height
x	_x
y	_y
visible	visible
enabled	enabled
color	_color

Working with Classes

CLASS

Properties:

```
width    =   300
height   =   120
x        =   300
y        =   200
visible  =   true
enabled  =   false
color    =   0xffff00
```

Dot Syntax

Working with Classes

CLASS

Properties:

```
var myInstance:Object = new Object();

myInstance.x = 300;
myInstance.y = 250;
myInstance.color = 0xff00dd;

myInstance.x = button3.x - 20;

myInstance.myValue = 30;
myInstance.myObject = new Object();
```

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

What a class can do.

for a Tiger

- run
- jump
- growl
- roar
- sleep

for a Sound

- load
- play
- stop
- setVolume
- setPan

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

What a class can do.

Collection of Functions()

function()

function(parameters)

Dot Syntax

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

```
var myTiger:Tiger = new Tiger();

myTiger.run();
myTiger.jump();
myTiger.roar();

run(speed:Number, distance:Number,
        direction:String);

myTiger.run(8);
myTiger.run(30, 5, “right”);
```

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

Events:

Trigger something to happen

Pick up on user input

All events need to be connected to a function.

Functions for Events are called Event Handlers

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

Events:

Button:

- click
- mouseOver
- mouseOut
- rollOver
- rollOut
- mouseDown
- mouseUp

Dot Syntax

Working with Classes

CLASS

Properties:

Methods:

Events:

ActionScript 1/2 and JavaScript:

```
myButton.click = function;
```

ActionScript 3:

```
myButton.addEventListener(
        MouseEvent.CLICK, function);
```

Dot Syntax