Hog Contest Rules

- Two people submit one entry; Max of one entry per person
- The score for an entry is the sum of win rates against every other entry.
- All strategies must be deterministic, pure functions of the current player scores! Non-deterministic strategies will be disqualified.
- To enter: submit proj1contest with a file hog.py that defines a final_strategy function by Monday 9/24 @ 11:59pm
- All winning entries will receive 2 points of extra credit
- The real prize: honor and glory

Fall 2011 Winners
Keegan Mann, Yan Duan & Ziming Li, Brian Prike & Zhenghao Qian, Parker Schuh & Robert Chatham

Choosing Names

Names typically don't matter for correctness  
but  
they matter tremendously for legibility

```python
>>> from operator import mul
def square(let):
    return mul(let, let)
```

Which Values Deserve a Name

Repeated compound expressions:
```python
if sqrt(square(a) + square(b)) > 1:
x = x + sqrt(square(a) + square(b))
```

```python
h = sqrt(square(a) + square(b))
if h > 1:
x = x + h
```

Meaningful parts of complex expressions:
```python
x = (-b + sqrt(square(b) - 4 * a * c)) / (2 * a)
d = sqrt(square(b) - 4 * a * c)
x = (-b + d) / (2 * a)
```

Test-Driven Development

Write the test of a function before you write the function

- A test will clarify the (one) job of the function
- Your tests can help identify tricky edge cases

Develop incrementally and test each piece before moving on

- You can't depend upon code that hasn't been tested
- Run your old tests again after you make new changes

Function Decorators

```python
@trace1
def triple(x):
    return 3 * x
```
Functional Abstractions

- Square takes one argument.
- Square has the intrinsic name square.
- Square computes the square of a number.
- Square computes the square by calling mul.

```python
def square(x):
    return mul(x, x)
```

What does `sum_squares` need to know about `square`?

- Square takes one argument. **Yes**
- Square has the intrinsic name square. **No**
- Square computes the square of a number. **Yes**
- Square computes the square by calling mul. **No**

```python
def square(x):
    return pow(x, 2)
def square(x):
    return mul(x, x-1) + x
```

If the name “square” were bound to a built-in function, `sum_squares` would still work identically.

---

Data

Student seating preferences at MIT

![Front of the classroom](http://www.skyrill.com/seatinghabits/)

---

Objects

- Representations of information
- Data and behavior, bundled together to create...

**Abstractions**

- Objects represent properties, interactions, & processes
- Object-oriented programming:
  - A metaphor for organizing large programs
  - Special syntax for implementing classic ideas

(Demo)

---

Native Data Types

In Python, every object has a type.

```python
>>> type(today)
<class 'datetime.date'>
```

Properties of native data types:

1. There are primitive expressions that evaluate to native objects of these types.
2. There are built-in functions, operators, and methods to manipulate these objects.

---

Python Objects

In Python, every value is an object.

- All objects have attributes
- A lot of data manipulation happens through methods
- Functions do one thing; objects do many related things

The next four weeks:

- Use built-in objects to introduce classic ideas
- Create our own objects using the built-in object system
- Implement an object system using built-in objects

---

Numeric Data Types

**Numeric types in Python:**

```python
>>> type(2)
<class 'int'>
```

Represents integers exactly

```python
>>> type(1.5)
<class 'float'>
```

Represents real numbers approximately

```python
>>> type(1+1j)
<class 'complex'>
```

(demo)
Care must be taken when computing with real numbers!

Representing real numbers:

False in a Boolean contexts:

1/3 = 0.3333333333333333

Moral of the Story

Life was better when numbers were just numbers!

Having to know the details of an abstraction:

- Makes programming harder and more knowledge-intensive
- Creates opportunities to make mistakes
- Introduces dependencies that prevent future changes

Coming Soon: Data Abstraction