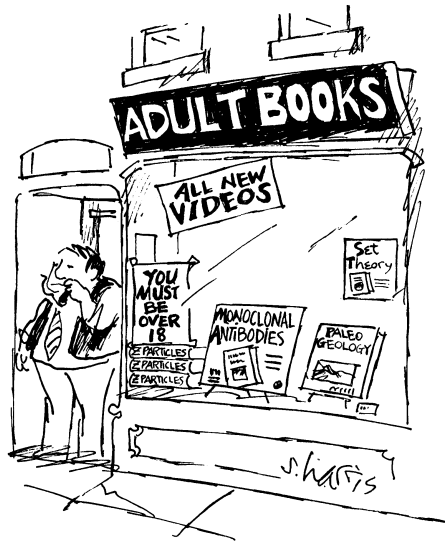


Sex, Religion, Politics, Java



One-Slide Summary

- **Languages** may change way we think.
- Java is verbose and **object-oriented**.
- An **object** packages state and procedures.
- A procedure on an object is called a **method**. We **invoke** a method by sending the object a **message**.
- **Inheritance** allows one object to refine and reuse the behavior of another. This is a good thing.
- To **evaluate** a name, **walk up** the frames until you find a definition.
- A **golden age** is a period when knowledge or quality increases rapidly.

#2

Why Learn New Languages?

- Languages change the way we think.
 - The **linguistic relativity principle** (also known as the **Sapir-Whorf Hypothesis**) is the idea that the varying cultural concepts and categories inherent in different languages affect the cognitive classification of the experienced world in such a way that speakers of different languages think and behave differently because of it. Roger Brown has drawn a distinction between weak linguistic relativity, where **language limits thought**, and strong linguistic relativity, where language determines thought. [Wikipedia]
- See also: Orwell's *1984*

#3

Why Learn New Languages?

- Deepening Understanding
 - By seeing how the same concepts we encountered in Scheme are implemented by a different language, you will **understand those concepts better** (especially procedures, assignment, and data abstraction).
- Building Confidence
 - By learning Java (mostly) on your own, the next time you encounter a problem that is best solved using a language you don't know, you will **be confident you can learn it** (rather than trying to use the wrong tool to solve the problem.)

#4

Why Learn New Languages

- Fun! Programming in can be Java is fun.
- Especially because:
 - It is commonly-used to solve real-world problems.
 - It is well-suited to group work.
 - It makes it easy to catch errors in advance.
 - It is strongly object-oriented.
 - They were going to name it "Oak" after the tree outside the office window, but that was already trademarked.

#5

Java

- **Java** is a **universal programming language**.
 - Everything you can compute in Python you can compute in Java, and vice versa
 - PS 7: implement a Python interpreter in Java
 - Chapter 12: more formal definition of a universal programming language
- Java is an **imperative language**.
 - Designed to support programming where most of the work is done using **assignment statements**
 - $x = \text{sqrt}(4) + 1;$

#6

Objectifying Java

- Java is also an **object-oriented language**.
 - **Objects** encapsulate **state** (i.e., variables and information) and the **methods** that operate on that state together.
 - In Java, almost all data are objects.
 - Problem Set 6 covers programming with objects.
 - Java has built-in support for classes, methods and inheritance.

#7

Learning New Languages

- **Syntax**: Where the {, !, (, :, etc., all go
 - If you can understand a BNF grammar, this is easy
 - But it still takes some getting used to
- **Semantics**: What does it mean?
 - Learning the evaluation rules
 - This is harder, but most programming languages have very similar rules (with subtle differences)
- **Style**: What are the idioms and customs?
 - Many years to be a “professional” Java programmer, but not long to write a program

#8

Java If

- **Instruction ::=**

```
if ( Expression ) {  
    Block  
} else {  
    Block  
}
```
- **Semantics**: Evaluate the *Expression* (which must be a Boolean). If it evaluates to a true value, evaluate the first *Block*. Otherwise, evaluate the second *Block*.
- You can omit `else { ... }`

#9

Java If Example

```
if (this_one > best_sofar) {  
    System.out.println(“This one is better!”);  
} else {  
    System.out.println(“Not better!”);  
}
```



Learning Java

- We will introduce (usually informally) Java constructs in class as we use them (and in example code in PS5 and PS6)
- The “Java Lab Guide” is a video introduction to Java and Eclipse:
 - Java : Eclipse :: Python : PyCharm
 - Covers what you need for PS5.
- On-line Java documentation

#11

Making Objects

```
ClassDefinition ::=  
public class Name {  
    FunctionOrFieldDefinitions  
}
```

```
public class Dog {  
    public static void bark() {  
        System.out.println(“wuff wuff wuff”);  
    }  
}
```

*In Washington, it's dog eat dog.
In academia, it's exactly the opposite.
- Robert Reich*

#12

Making a Dog

Expression ::= new ClassName(args)

```
public class Dog {
    public static void bark() {
        System.out.println("wuff wuff wuff");
    }
}
```

...

```
Dog spot = new Dog();
```

In Java, you must *declare* a variable with its type before you give it a value.

#13

Java Procedures (= Methods)

```
public class Dog {
    public static void bark() {
        System.out.println("wuff wuff wuff");
    }
}
```

MethodDefinition ::= Modifiers Type Name (Params) Block
Params ::= SomeParams | epsilon
SomeParams ::= Type Name | Type Name, SomeParams
Block ::= { Statements }
Statements ::= Statement ; MoreStatements
MoreStatements ::= epsilon
 | Statement ; MoreStatements

#14

Some Java Procedures

```
int square(int x) {
    return x*x;
}

int bigger(int a, int b) {
    if (a > b)
        return a;
    else return b;
}
```

```
int biggest(int [] lst) {
    int biggest = lst[0];
    for (int i = 1; i < lst.length; i = i + 1)
        if (lst[i] > biggest) biggest = lst[i];
    return biggest;
}
```

#15

Barking: Invoking Methods

ApplicationStmt ::= Expr.Name(Args)
Args ::= epsilon | MoreArgs
MoreArgs ::= Argument , MoreArgs
 | Argument

```
public class Dog {
    public static void bark() {
        System.out.println("wuff wuff wuff");
    }
}
```

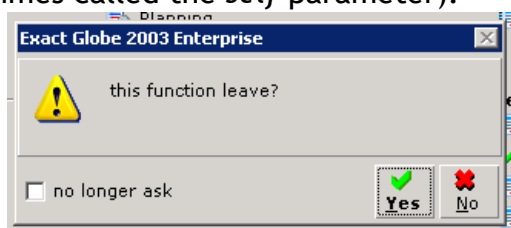
...

```
Dog spot = new Dog();
spot.bark(); // Invoke bark on spot
wuff wuff wuff wuff
```

#16

Object Lingo

- “Apply a procedure” = “Invoke a method”
- We apply a procedure to parameters.
- We invoke a method on an object, and pass in parameters.
 - Inside a method you can also see the object itself (sometimes called the *self* parameter).



#17

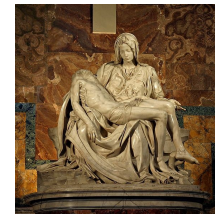
Liberal Arts Trivia: Art History and American Literature

- Give the Renaissance master (or Ninja Turtle) associated with each work of art:



(a) Tomb of Antipope John XXIII

(b) Mona Lisa



(c) Pieta

(d) Transfiguration



#18

Liberal Arts Trivia: Polish History, Chemistry, and Physics

- This physicist and chemist of Polish upbringing and French citizenship was the first person honored with two Nobel prizes, the first woman to win a Nobel prize, and the first woman to serve as a professor at the University of Paris. The world's first studies into the treatment of cancers using radioactive isotopes were conducted under her direction.

#19

Liberal Arts Trivia: Cooking

- This Japanese delicacy is vinegared rice, usually topped with other ingredients, including fish. The dish as we know it today was invented as a fast food by Hanaya Yohei at the end of the Edo period (19th century) in Tokyo: it could be eaten on the road side or in a theatre using fingers or chopsticks. The basic idea can be traced back to 4th century BCE China as a preservative: the fermentation of the rice prevents the fish from spoiling.

#20

Dogs with Names



#21

Review: Making a Dog

```
public class Dog {
    public String name; // Field (= State)
    public Dog(String n) { // Constructor
        name = n; // Initialize Field
    }
    public void bark() { // Method
        println(name + "says wuff!");
    } // Methods can see fields (like name)!
    ...
    Dog myDog = new Dog("Spoticus"); // "new" calls Constructor, returns new object
    myDog.bark(); // Invoke Method
    Spoticus says wuff!
    Dog yourDog = new Dog("Ginger"); // Not all objects have the same state!
    yourDog.bark();
    Ginger says wuff!
```

```
public class Dog {
    Public void bark() {
        System.out.println("wuff wuff");
    }
    ...
    Dog spot = new Dog(); // spot has type Dog
    You must declare the type first!
```

#22

Java "Lists"

- Python has a built-in datatype for a list of fixed length. It is called "array", written [].
- ```
int [] myArray = {8,6,7};
println(myArray[0]);
1
println(myArray.length);
3
String [] myNames = {"Wes", "Weimer"};
println(myNames[1]);
Weimer
```

#23

## Implementing square\_each in Java

```
def square_each(lst):
 for i in range(len(lst)):
 lst[i] = lst[i] * lst[i] # imperative!
```

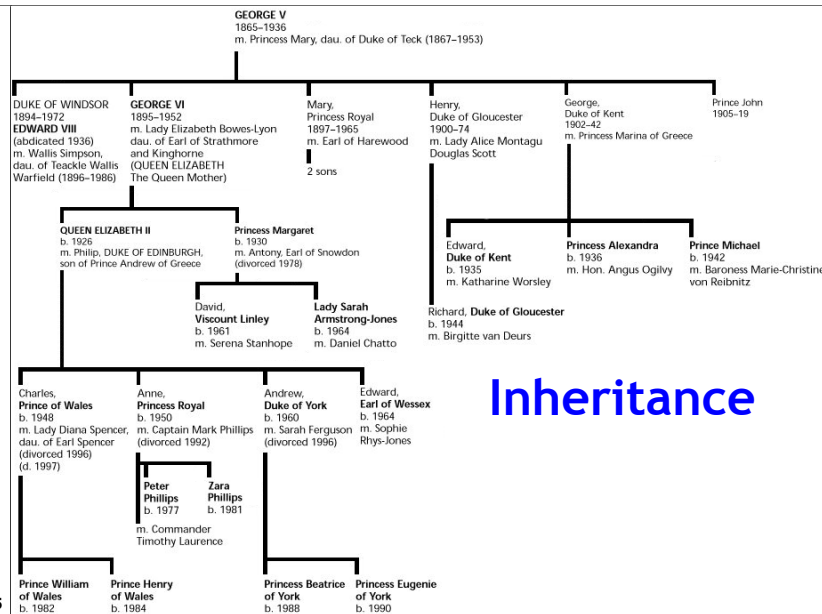
- Let's do a literal translation of this into Java.



## Java square\_each

```
public static void square_each(int [] lst) {
 for (int i = 0; i < lst.length; i = i + 1) {
 lst[i] = lst[i] * lst[i] ; // still imperative!
 }
}
```

- Just like the previous one, this mutates lst.



#25

## Hey, Scooby!



```
public class Dog {
 public Dog(n) { name = n; }
 public String name;
 public void bark() {
 println("wuff wuff");
 }
}

public class TalkingDog extends Dog {
 public void speak(String words) {
 println(name + " says " + words);
 }
} // inherits all Dog fields and methods

TalkingDog scooby = new TalkingDog("Scooby");
scooby.speak("solve the mystery!");
Scooby says solve the mystery!
```

#27

## Subclasses

```
public class TalkingDog extends Dog {
 public void speak(String words) {
 println(name + " says " + words);
 }
} // inherits all Dog fields and methods
```

- TalkingDog is a **subclass** of Dog.
- Dog is the **superclass** of TalkingDog.
  - Every TalkingDog is also a Dog.
  - (But not vice-versa.)

#28

## Every Dog Has Its Day

```
public class Dog {
 public Dog(n) { name = n; }
 public String name;
 public void bark() {
 println("wuff wuff");
 }
}

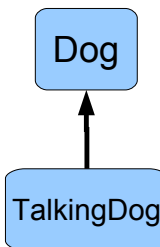
public class TalkingDog extends Dog {
 public void speak(String words) {
 println(name + " says " + words);
 }
} // inherits all Dog fields and methods
```

```
Dog ginger = new Dog("Ginger");
TalkingDog scooby = new Dog("Scooby");
scooby.speak("snack!");
Scooby says snack!
ginger.speak("this won't work");
Type Error
scooby.bark();
wuff wuff
```

#29

## Speaking About Inheritance

- Inheritance is using the definition of one class to define another class.
- TalkingDog **inherits** from Dog.
- TalkingDog is a **subclass** of Dog.
- The **superclass** of TalkingDog is Dog.
- *These all mean the same thing!*



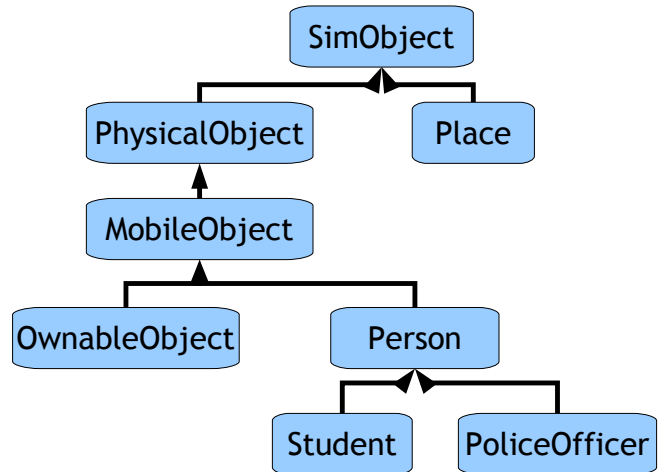
#30

## Problem Set 6

- Make an adventure game by programming with objects.
- Many objects in our game have similar properties and behaviors, so we use inheritance.

#31

## PS6 Classes



#32

## Object-Oriented Review

- An **object** is an entity that packages state and procedures.
- The state variables that are part of an object are called **instance variables**.
- The procedures that are part of an object are called **methods**.
- We **invoke** (call) a method. The object itself and its fields are also visible in a method.
- **Inheritance** allows one class to refine and reuse the behavior of another.
- A **constructor** is a procedure that creates new objects (e.g., `public Dog() { ... }`).

#33

## Golden Ages or Golden Catastrophes?

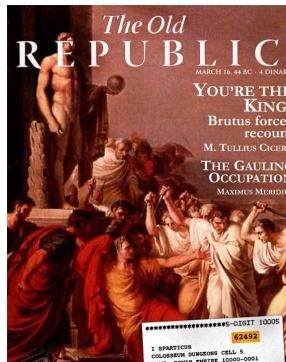


## The Real Golden Rule?

Why do fields like astrophysics, medicine, biology and computer science have “endless golden ages”, but fields like ...

- rock n' roll (1962-1973, or whatever was popular when you were 16)
- music (1775-1825)
- philosophy (400BC-350BC?)
- art (1875-1925?)
- soccer (1950-1966)
- baseball (1925-1950?)
- movies (1920-1940?)

have short golden ages?



## Malthusian Catastrophe

Reverend Thomas Robert Malthus, *Essay on the Principle of Population*, 1798

“The great and unlooked for discoveries that have taken place of late years in natural philosophy, the increasing diffusion of general knowledge from the extension of the art of printing, the ardent and unshackled spirit of inquiry that prevails throughout the lettered and even unlettered world, ... have all concurred to lead many able men into the opinion that we were touching on a period big with the most important changes, changes that would in some measure be decisive of the future fate of mankind.”



Source: The Warren J. Samuels Portrait Collection at Duke University

## Malthus' Postulates

"I think I may fairly make two postulata.

- First, that food is necessary to the existence of man.
- Secondly, that the passion between the sexes is necessary and will remain nearly in its present state.

These two laws, ever since we have had any knowledge of mankind, appear to have been fixed laws of our nature, and, as we have not hitherto seen any alteration in them, we have no right to conclude that they will ever cease to be what they now are..."

## Malthus' Conclusion

"Assuming then my postulata as granted, I say, that the power of population is indefinitely greater than the power in the earth to produce subsistence for man.

Population, when unchecked, increases in a geometrical ratio. Subsistence increases only in an arithmetical ratio. A slight acquaintance with numbers will show the immensity of the first power in comparison of the second."

## Malthusian Catastrophe

- Population growth is geometric:  $\Theta(k^n)$  ( $k > 1$ )
- Food supply growth is linear:  $\Theta(n)$

What does this mean as  $n \rightarrow \infty$ ?

$$\text{Food per person} = \text{food supply} / \text{population} \\ = \Theta(n) / \Theta(k^n)$$

As  $n$  approaches infinity, food per person approaches zero!

## Liberal Arts Trivia: American Studies

- This American social activist and leading figure of the woman's movement crafted the Declaration of Sentiments. Its presentation, at the first women's rights convention in Seneca Falls, is often credited with initiating the first woman's suffrage movement in the USA. Beyond voting rights, her work addressed parental and custody rights, employment and income rights, property rights, divorce laws, and birth control.

## Liberal Arts Trivia: Classics and Drama

- This ancient Greek tragedian playwright wrote Ajax, **Antigone**, Trachinian Women, **Oedipus the King**, Electra, Philoctetes and Oedipus at Colonus. He influenced the development of the drama by adding a third actor (reducing the importance of the chorus in the presentation of the plot) and putting a greater emphasis on character development.

## Malthus' Fallacy



## Malthus' Fallacy

He forgot how he started:

“The great and unlooked for discoveries that have taken place of late years in natural philosophy, the increasing diffusion of general knowledge from the extension of the art of printing, the ardent and unshackled spirit of inquiry that prevails throughout the lettered and even unlettered world...”

## Golden Age of Food Production

- Agriculture is an “endless golden age” field: production from the same land increases as  $\sim \Theta(1.02^n)$
- Increasing knowledge of farming, weather forecasting, plant domestication, genetic engineering, pest repellants, distribution channels, etc.

## Growing Corn

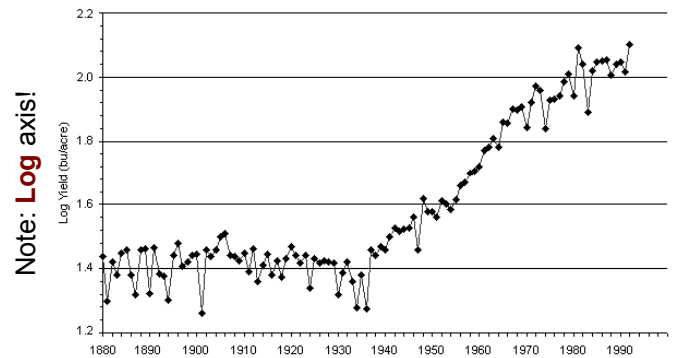


1906: < 1,000  
pounds per acre

2006: 10,000  
pounds per acre

Michael Pollan's *The Omnivore's Dilemma*

## Corn Yield



<http://www.agbioforum.org/v2n1/v2n1a10-ruttan.htm>

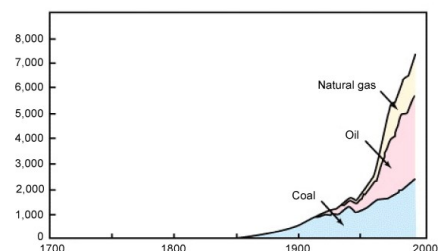
## Example: Norman Borlaug

- Father of the Green Revolution
  - Nobel Peace Prize, Presidential Medal of Freedom, Congressional Gold Medal (one of five to win all three), India's Padma Vibhushan
- "At a time when doom-sayers were hopping around saying everyone was going to starve, Norman was working. He moved to Mexico and lived among the people there until he figured out how to improve the output of the farmers. So that saved a million lives. Then he packed up his family and moved to India, where in spite of a war with Pakistan, he managed to introduce new wheat strains that quadrupled their food output. So that saved another million. You get it? But he wasn't done. He did the same thing with a new rice in China. He's doing the same thing in Africa -- as much of Africa as he's allowed to visit. **When he won the Nobel Prize in 1970, they said he had saved a billion people. That's BILLION! BUH!** That's Carl Sagan BILLION with a "B"! And most of them were a different race from him. **Norman is the greatest human being, and you probably never heard of him.**"
  - Penn Jillette, on the show Penn & Teller

## Upcoming Malthusian Catastrophes?

- Human consumption of fossil fuels grows as  $\Theta(k^n)$  (fairly large  $k$  like 1.08?)
- Available fuel is constant (?)

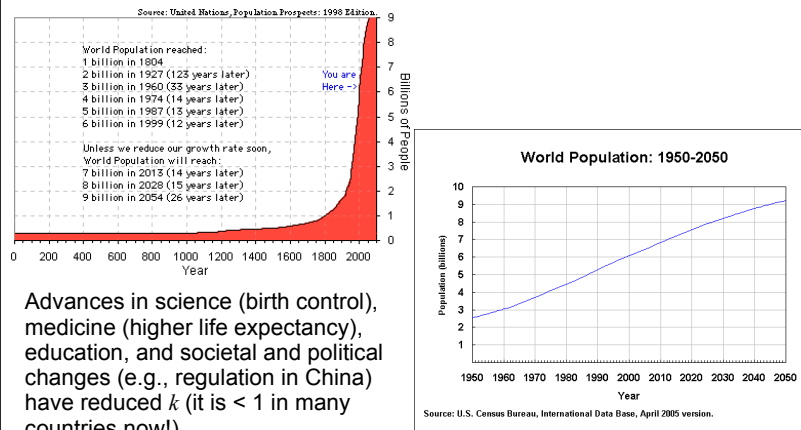
Fig. 3: Trends in World Fossil Fuel Consumption (Million tons oil equivalent)



Source: Environment Agency's "White Paper on the Environment" (1998)  
[http://www.wpp.mext.go.jp/hakusyo/book/hpag200001/hpag200001\\_2\\_006.html](http://www.wpp.mext.go.jp/hakusyo/book/hpag200001/hpag200001_2_006.html)



## Malthus was wrong about #2 Also



## “Cornucopian View”

- Few resources are really finite
- All scientific things seem to have endless golden ages
- (We hope) Human ingenuity and economics and politics will solve problems before they become catastrophes
  - No one will sell the last gallon of gas for \$4.00

## “Kay”-sian View

The best way to predict the future is to invent it.  
— Alan Kay

## Liberal Arts Trivia: French Literature

- This 19<sup>th</sup> century French writer and political activist was an exponent of the Romantic movement in France. Two of his volumes of poetry, *Les Contemplations* and *La Légende des siècles* are particularly critically acclaimed, and he is sometimes called the greatest French poet. Outside of France he is perhaps best known for *Les Misérables* and *Notre-Dame de Paris*.
- Bonus points: Give Valjean's prisoner number.

## Liberal Arts Trivia: French History



- This 1806 Parisian monument commemorates those who fought for France, particularly in the Napoleonic Wars. Underneath it is the Tomb of the Unknown Soldier from WWI.

## Liberal Arts Trivia: Mathematics

- This is a major area of mathematics that combines developments and concepts from set theory and geometry, such as those of dimension, space, transformation and shape. Of particular importance to this field are homeomorphisms, which can be viewed as continuous functions with continuous inverses. Subfields include point-set, algebraic, and geometric.

# Review: Names, Places, Mutation

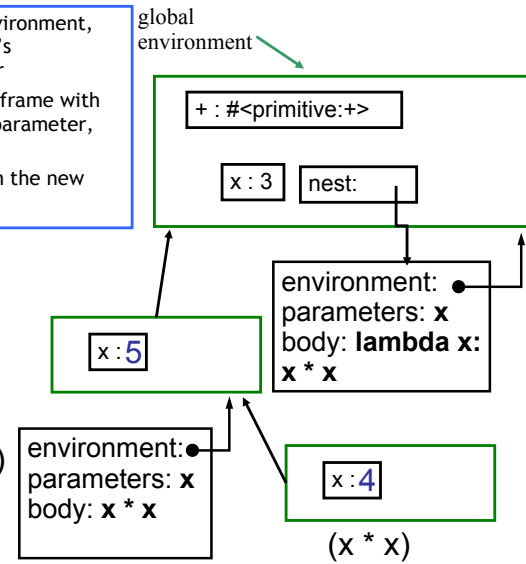
- A **name** is a **place** for storing a value.
- A **definition** creates a new place.
- **[1,2]** application creates two new places, the **[0]** and the **[1:]**.
- A **frame** is a collection of places.
- An **environment** is a frame and a pointer to a parent environment.
- The **global environment** has no parent.
- **name = expr** changes the value in the place **name** to the value of **expr**.

1. Construct a new environment, parent is procedure's environment pointer
2. Make places in that frame with the names of each parameter, and operand values
3. Evaluate the body in the new environment

```

>>> x = 3
>>> nest =
lambda x :
lambda x : x * x
>>> (nest(5))(4)
16

```



#56

## Charge



- When picking majors, pick a short golden age field that is about to enter its short golden age
  - This requires vision and luck!
- Play it safe by picking an endless golden age field (CS is a good choice for this!)
- Start PS6 early
- Start thinking about PS9 Project ideas
  - If you want to do an “extra ambitious” project convince me your idea is worthy before Nov 10 (ps7 and 8) / Nov 17 (ps8)

## Homework

- PS 5 due *today*
- PS 6 due *soon*

#58