CSCI 334:
Principles of Programming Languages

Lecture 21: Object-oriented programming

Instructor: Dan Barowy

Williams

Topics

Programming in the large/small
Object-oriented programming
Dynamic dispatch

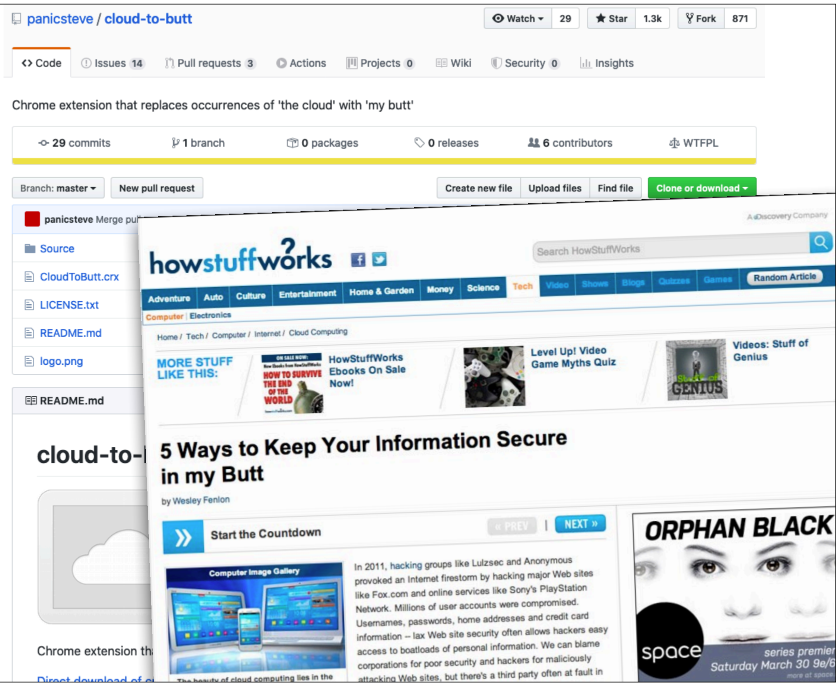
Your to-dos

1. Lab #10, **due Sunday 12/3**
2. Want to talk about your project?
   Office hours tomorrow 10-11am, 1:30-2:30pm.

Final project timeline

1. ~~Project proposal (Lab 8), **due Sun 11/12**~~
2. ~~Minimally working version (Lab 9), **due Sun 11**~~**/19**
3. Language specification doc (Lab 10), **due Sun 12/3**
4. Mostly working version (Lab 11), **due Sun 12/10**
5. Project + video presentation (Lab 12), **due Sun 12/17**

Programming in the small



Chrome extension that replaces occurrences of 'the cloud' with 'my butt'

Chrome extension th

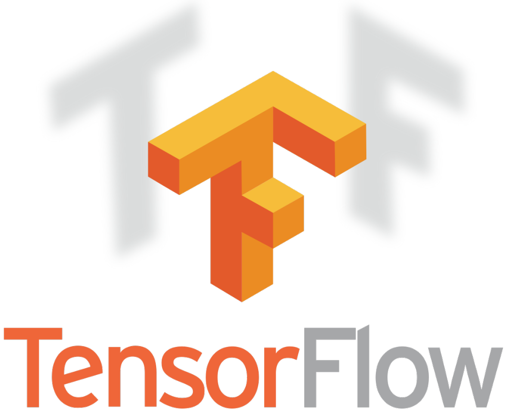
5 Ways to Keep Your Information Secure in my Butt

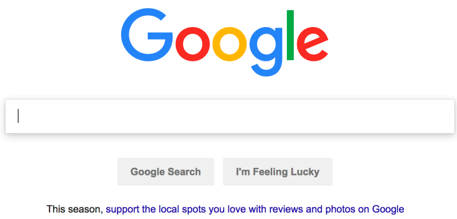
In 2011, hacking groups like Lulzsec and Anonymous provoked an Internet firestorm by hacking major Web sites like Fox.com and online services like Sony's PlayStation Network. Millions of user accounts were compromised. Usernames, passwords, home addresses and credit card information -- lax Web site security often allows hackers easy access to boatloads of personal information. We can blame corporations for poor security and hackers for maliciously attacking Web sites, but there's a third party often at fault in

The beauty of cloud computing lies in the

Programming in the large

Ruby on Rails
Sustainable productivity for web-application development


TensorFlow



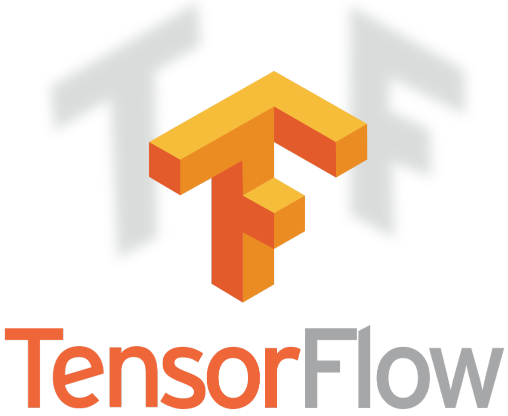
This season, support the local spots you love with reviews and photos on Google

What languages?

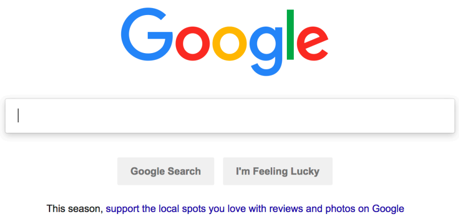
Java

C++

Ruby



C++, Python



This season, support the local spots you love with reviews and photos on Google

Java

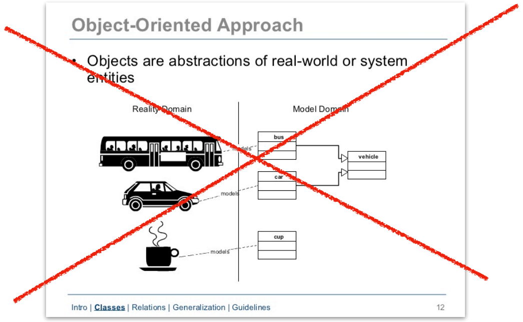
Object-Oriented Programming

Object-Oriented Programming

- OOP is both a **language design** and a **way of programming** (OO design).
- OOP is possibly the **most impactful** development in the history of programming languages.

What OOP is Not

- Many, many instructors introduce OOP as a way of naturally simulating the world.



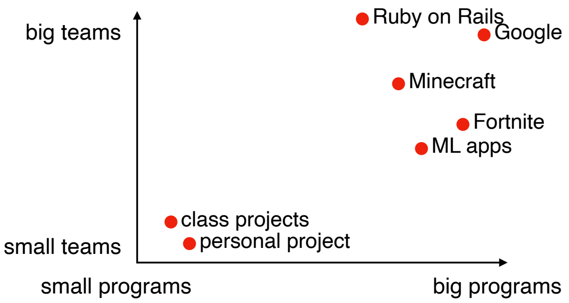
Object-Oriented Approach

- Objects are abstractions of real-world or system entities

- This view **entirely misses the point** of OOP!

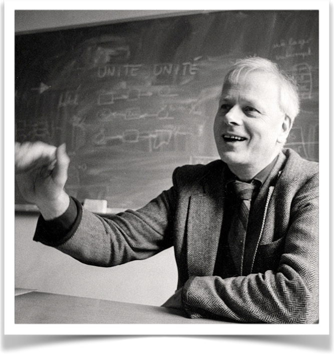
What OOP is

- Object-oriented programming is actually about **scalability**.
- The original motivation was motivated by two questions:
  - How do we manage **big codebases**?
  - How do big teams of programers **collaborate effectively**?



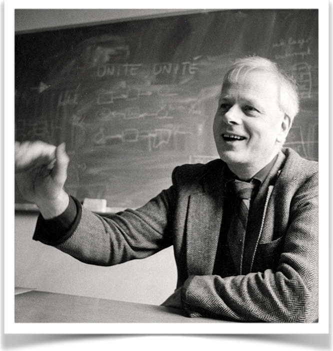
History

- First language recognizable as OO: Simula-67.
- Developed by Kristen Nygaard and others at the Norwegian Computing Center.
- Grew out of frustrations using ALGOL.
- Original plan was to add an “object” library, inspired by C.A.R. Hoare’s “record classes”.
- It was eventually realized that a fundamentally different way of structuring a program was possible; Simula became its own language.

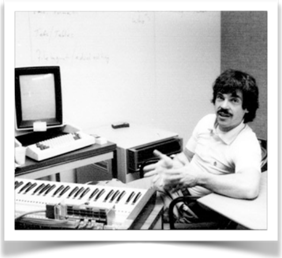


History

- But Simula-67 was not the most influential OO language.
- That language was…



Smalltalk



Alan Kay
Essentially invented the laptop/tablet ("Dynabook")

Turing Award

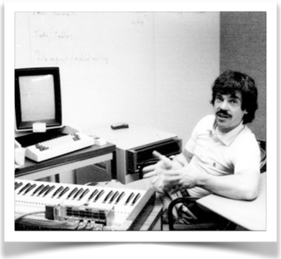
Dan Ingalls
Essentially invented object oriented programming

Grace Murray Hopper Award

Adele Goldberg
Essentially invented graphical user interfaces

ACM Software Systems Award

Smalltalk

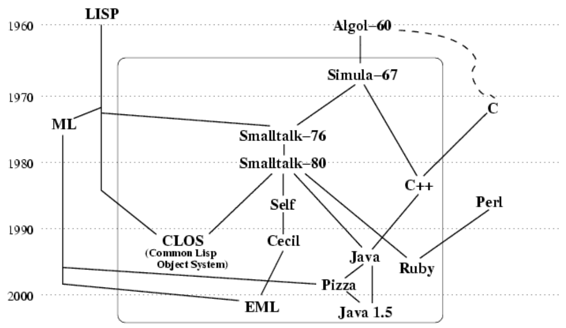


- First mainstream OO success: Smalltalk
- Developed by Alan Kay, Dan Ingalls, and Adele Goldberg at Xerox PARC and later Apple Computer.
- Used to implement major components of the groundbreaking Xerox Alto computer: OS, compiler, GUI, applications.
- Highly influential. E.g., C++, Java, Ruby, etc.

Smalltalk

And they showed me really three things. But I was so blinded by the first one I didn't even really see the other two. One of the things they showed me was object orienting programming. They showed me that but I didn't even see that. The other one they showed me was a networked computer system… they had over a hundred Alto computers all networked using email etc., etc. I didn't even see that. I was so blinded by the first thing they showed me which was the graphical user interface… within you know ten minutes it was obvious to me that all computers would work like this some day.

Smalltalk



OK, really, what is OO?

Object-oriented programming is composed primarily of four key language features:

1. Abstraction
2. Dynamic dispatch
3. Subtyping
4. Inheritance

Purpose: polymorphism at scale

OK, really, what is OO?

Object-oriented programming is composed primarily of four key language features:

1. Abstraction
2. **Dynamic dispatch**
3. Subtyping
4. Inheritance

In my mind, this is OO’s killer feature.

Purpose: polymorphism at scale

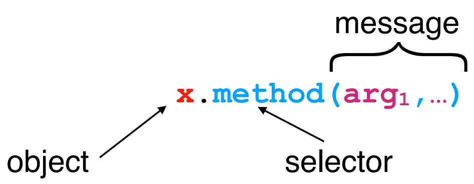
“Object-oriented programming is a solution to complexity”

Dynamic Dispatch

(the secret to understanding how
Java, Python, Ruby, etc. work)

Dynamic Dispatch

- **Dynamic dispatch** is the OO mechanism for **polymorphism**.
- Functions ("**methods**") are always bound to an object (or class)
- A method is called ("**dispatched**") by sending a "**message**" to the "**selector**" of an object.



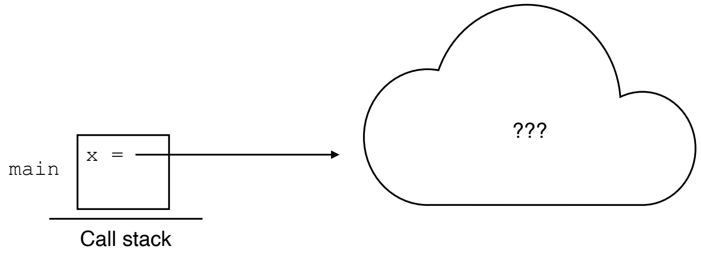
Dynamic Dispatch

- Suppose we have:

```
class Number {
  int value;

  public Number(int v) {
    value = v;
  }

  public int getValue() {
    return value;
  }

  public String squee() {
    return “squee!”;
  }
}
```

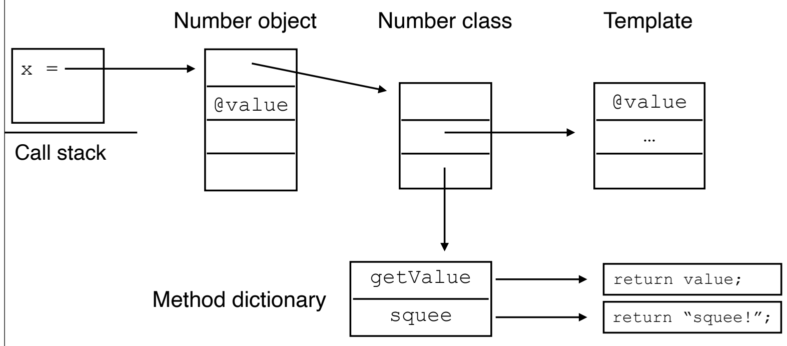
Dynamic Dispatch

- `x` is a `Number`.
- How does an object work?

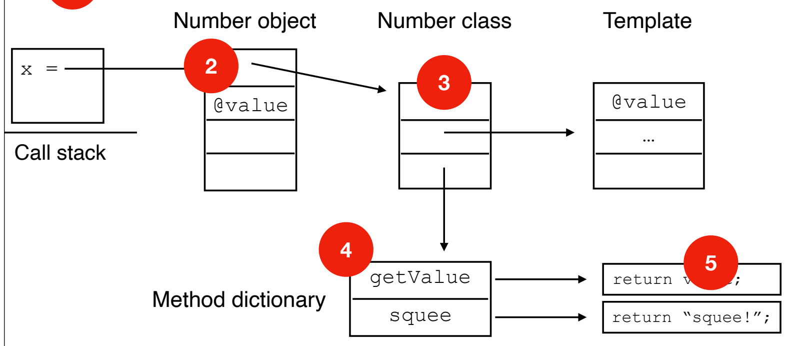


Dynamic Dispatch

- Dynamic dispatch is an **algorithm** for finding the **implementation** of a given **selector** (i.e., method).



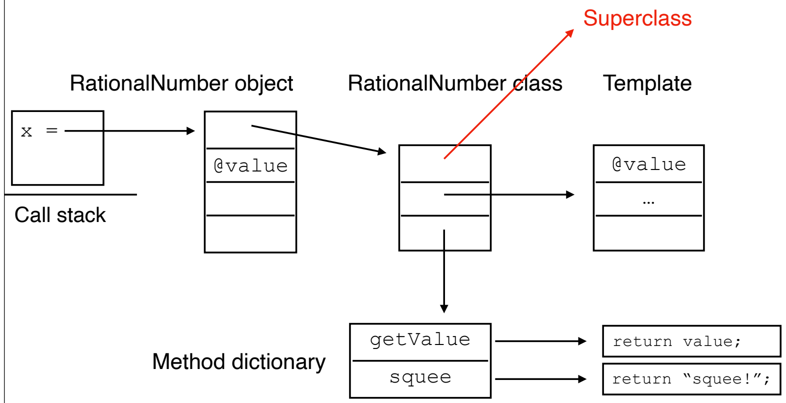
1. Call `x.getValue`
2. `x.getValue` message dispatched to `x`
3. `x.getValue` message forwarded to `Number`
4. `x.getValue` message lookup in method dictionary
5. `x.getValue` executed.



Object-oriented programming is a solution to complexity

Inheritance

- One small change enables inheritance.



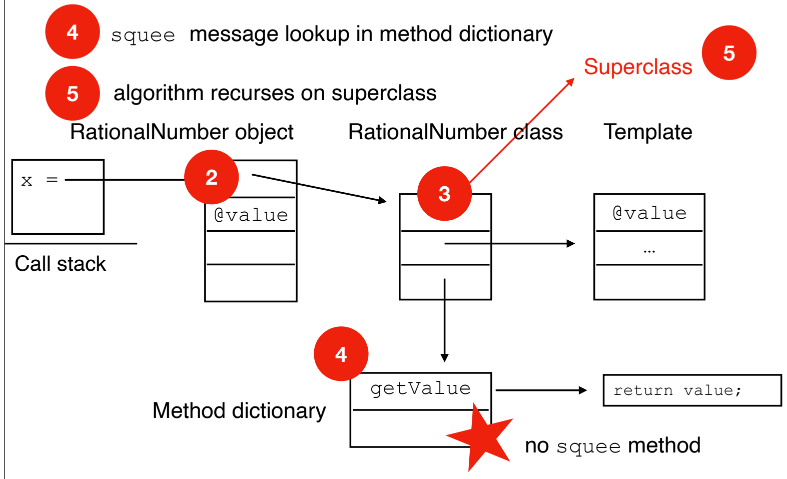
1 Call `x.squee`

2 `squee` message dispatched to `x`

3 `squee` message forwarded to `RationalNumber`

4 `squee` message lookup in method dictionary

5 algorithm recurses on superclass



Object-oriented programming is a solution to complexity

Recap & Next Class

This lecture:

OOP

Next lecture:

Student Course Surveys

How to give a good technical talk