CSCI 334: Principles of Programming Languages

Lecture 22: How to give a good talk / OOP

Instructor: Dan Barowy

Williams

Topics

How to give a good talk

FlashRelate (2015)

Object-Oriented Programming

Your to-dos

- 1. Final project, due Wednesday 5/14
- Want to talk about your project?
 Office hours today 4-6pm, tomorrow 3-5:30pm, TA hours until Wed night.
- 3. Resubmissions due: May 20. See instructions in handout.

(Don't forget to fill out the Google Form!)

4. Note: with (2 exams + 8 quizzes) x 63 students, there was a lot of grading, and it is likely that I made at least one mistake. If you think I made one, feel free to talk to me in person!

Announcements



Ward Prize Talks Friday, May 16 @ 2:35pm Wege Auditorium

Presentations by students nominated for the 2025 Rich Ward Prize for Best Student Project in Computer Science. CS colloquium credit for attendance!

End-of-year celebration to follow in Science Quad

CSCI 334: Principles of Programming Languages

> Lecture 22-1: How to give a **good** talk

> > Instructor: Dan Barowy

Williams

How to give a good talk

Five tips

One: Have a story





Three: Don't make your audience read

Four: Show by example



Five: Stay on script





Not a bad idea to finish with a concluding slide

https://infrastructor.tools

Daniel W. Barowy and William K. Jannen. 2020. Infrastructor: Flexible, No-Infrastructure Tools for Scaling CS. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20).*

Contributions welcome!



Williams





CHECKCELL: Data-Debugging for Spreadsheets



UMassAmherst

- Introduces data-debugging.
- Works as long as there are formulas.
- Accurate.
- Fast.
- Available to download now:
 <u>https://checkcell.org</u>

CSCI 334: Principles of Programming Languages

Thanks!

Hopefully I gave a good talk

Instructor: Dan Barowy

Williams

Video presentation

The Heilmeier Catechism



DARPA operates on the principle that generating big rewards requires taking big risks. But how does the Agency determine what risks are worth taking?

George H. Heilmeier, a former DARPA director (1975-1977), crafted a set of questions known as the "Heilmeier Catechism" to help Agency officials think through and evaluate proposed research programs.

What are you trying to do? Articulate your objectives using absolutely no jargon.

How is it done today, and what are the limits of current practice?

What is new in your approach and why do you think it will be successful

Who cares? If you are successful, what difference will it make?

https://www.darpa.mil/work-with-us/heilmeier-catechism

PL can do magic things...

FlashM demo

Let's step back a little...

Programming in the small















	Go	ogle		
1				
	Google Search	I'm Feeling Lucky		
This season,	support the local spots you	love with reviews and ph	otos on Google	

What languages?





C++





	Go	ogle		
1				
	Google Search	I'm Feeling Lucky		
This season, sup	port the local spots yo	u love with reviews and photos on (Google	
	Ja	iva		

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 "simulating the world."



• This view entirely misses the point of OOP!



"Object-oriented programming is a solution to complexity"

-Dan Ingalls, inventor of the SmallTalk language

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? Big teams Pluby on Rails Google Minecraft Fortnite machine learning It was an all teams personal project personal project big programs

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.





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

One of the things they showed me was object orienting programming [...] within you know ten minutes it was obvious to me that all computers would work like this some day.



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



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

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

This is the killer

feature.

Purpose: polymorphism at scale

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 **method**.







"Object-oriented programming is a solution to complexity"

(video)

Recap & Next Class

This lecture:

OOP Inheritance

How to give a good talk

Next lecture:

No next lecture!