Penn Engineering
University of Pennsylvania

On Building a Successful Research Group

Stephanie Weirich
University of Pennsylvania

WHAT THIS TALK IS NOT ABOUT: how to fund and lead a large research lab, with 2-3 postdocs, 8-15 PhD students, and a bunch of undergrads. I don’t know how to do that.

WHAT THIS TALK IS ABOUT: how to create an environment where professors, postdocs, PhD students, undergraduates can thrive together.

PL Club and friends

May 11, 2018

What is PL Club?

- Benjamin Pierce, Steve Zdancewic, Stephanie Weirich and friends
- Group of like-minded faculty, students, and researchers interested in *programming languages* (broadly construed)
- Venue for research talks & practice presentations
- *Sometimes* reading group
- Host for visitors
- Social organization for fun activities

PL Club Highlights

2002	Steve and I join Benjamin Pierce at Penn
2000-2004	ICFP Programming Competitions
2005	POPLMark Challenge
2007	Software Foundations textbook
2010	CIS 120 Redesign
2017	The Science of Deep Specifications
2020	Virtual ICFP
2023	REPL program

ICFP Programming Contest

ICFP Programming Contest

2004 – Hosts!

Programming Languages Research at Penn

People

- Benjamin Pierce
- Stephanie Weirich
- Steve Zdancewic

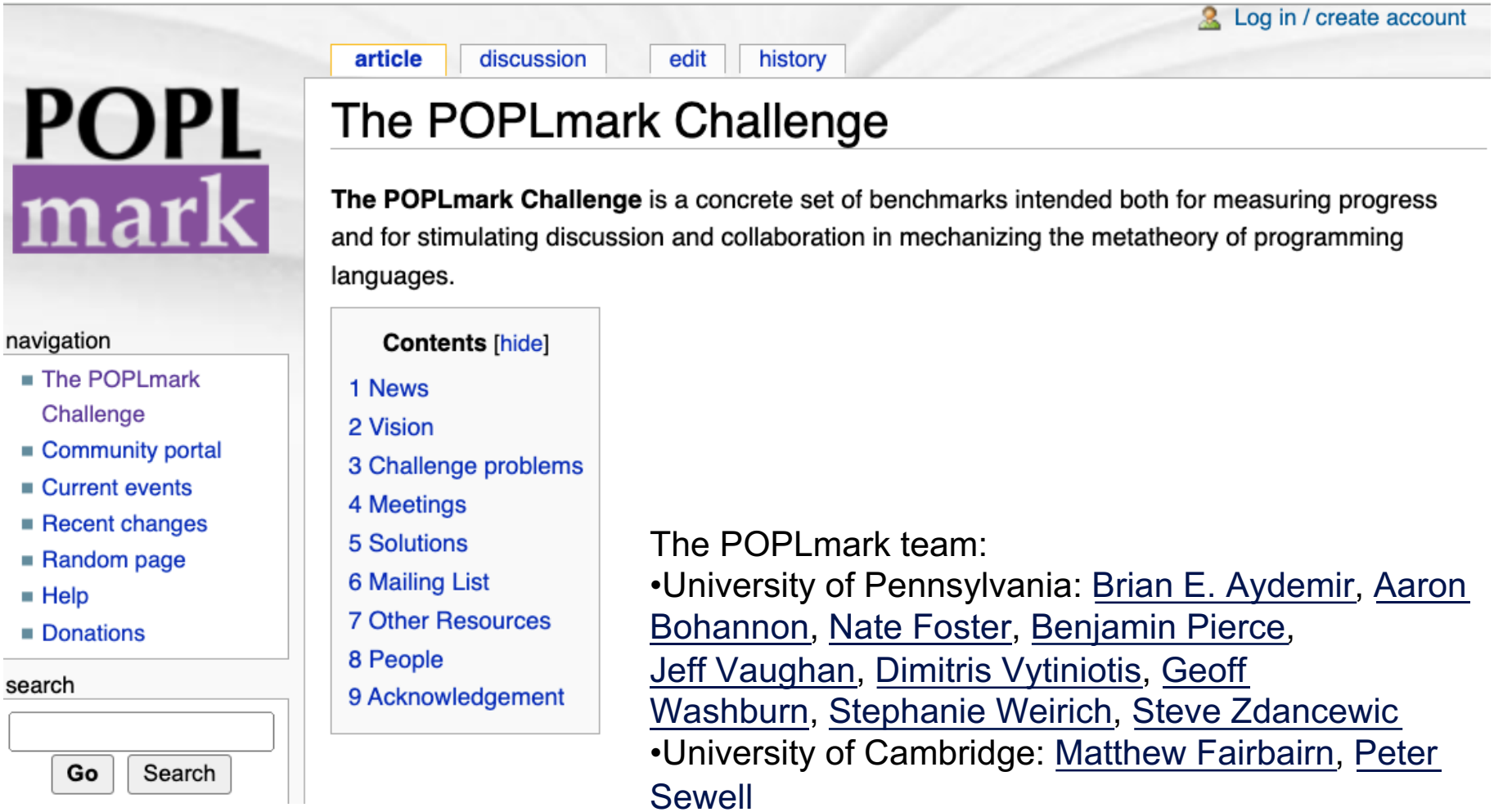
Research Projects

- Synchronization
  - Unison
  - Harmony
- Language support for XML
  - XDuce
  - Xtatic
- Security-oriented languages
- Type-directed programming
  - Extensible intensional type analysis
  - Type-directed Java
- POPLmark (formerly Mechanized Metatheory for the Masses!)

Publications

- All PLClub publications

Seminars and Courses



The POPLmark team:

•University of Pennsylvania: Brian E. Aydemir, Aaron Bohannon, Nate Foster, Benjamin Pierce, Jeff Vaughan, Dimitris Vytiniotis, Geoff Washburn, Stephanie Weirich, Steve Zdancewic

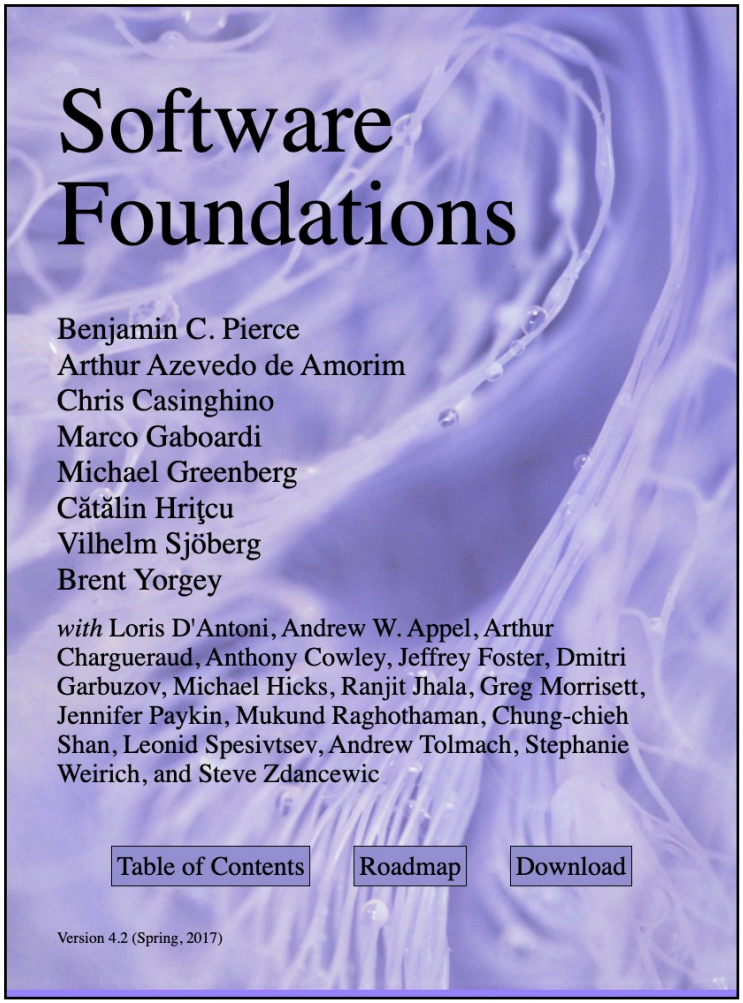
•University of Cambridge: Matthew Fairbairn, Peter Sewell

POPLmark Challenge (2005)

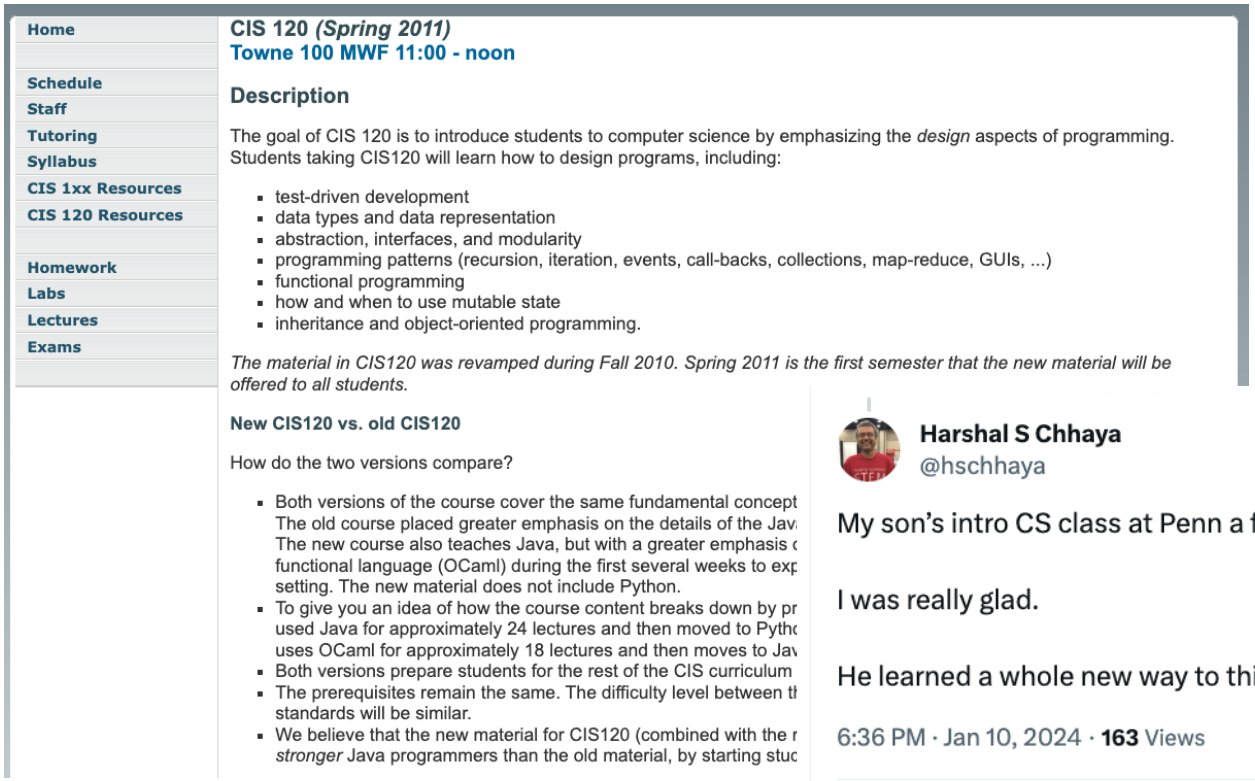
- Brian E. Aydemir, Aaron Bohannon, Matthew Fairbairn, J. Nathan Foster, Benjamin C. Pierce, Peter Sewell, Dimitrios Vytiniotis, Geoffrey Washburn, Stephanie Weirich, and Steve Zdancewic. **Mechanized Metatheory for the Masses: The POPLmark Challenge**. In *The 18th International Conference on Theorem Proving in Higher Order Logics (TPHOLs)*, pages 50--65, Oxford, UK, August 2005.
- Workshop on Mechanizing Metatheory (2006-2010)
- POPL 2008 tutorial
- Brian Aydemir, Arthur Charguéraud, Benjamin C. Pierce, Randy Pollack, and Stephanie Weirich. **Engineering Formal Metatheory**. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 3--15, January 2008.
- Benjamin C. Pierce and Stephanie Weirich. **Preface to Special Issue on the POPLMark Challenge**. *J. Autom. Reasoning*, 49(3):301--302, 2012.

Software Foundations (2

```
(** * Software Foundations, Formally
      Benjamin C. Pierce
      Version of 9/4/2007

   Before handing in this file with your homework solutions,
   please fill in the names of all members of your group:

     FILL IN HERE

*)

(* ================================================================= *)
(** * LECTURE 1 *)

(** This file develops basic concepts of functional programming,
    logic, operational semantics, lambda-calculus, and
    static type systems, using the Coq proof assistant.  It
    is intended to be "read" in an interactive session with
    Coq under a development environment -- either CoqIDE or
    Proof General.
*)
```



CIS 120 Redesign (Fall 2010)



Home

Schedule
Staff
Tutoring
Syllabus
CIS 1xx Resources
CIS 120 Resources

Homework
Labs
Lectures
Exams

CIS 120 *(Spring 2011)*
Towne 100 MWF 11:00 - noon

Description

The goal of CIS 120 is to introduce students to computer science by emphasizing the *design* aspects of programming. Students taking CIS120 will learn how to design programs, including:

- test-driven development
- data types and data representation
- abstraction, interfaces, and modularity
- programming patterns (recursion, iteration, events, call-backs, collections, map-reduce, GUIs, ...)
- functional programming
- how and when to use mutable state
- inheritance and object-oriented programming.

The material in CIS120 was revamped during Fall 2010. Spring 2011 is the first semester that the new material will be offered to all students.

New CIS120 vs. old CIS120

How do the two versions compare?

- Both versions of the course cover the same fundamental concept
  The old course placed greater emphasis on the details of the Jav
  The new course also teaches Java, but with a greater emphasis c
  functional language (OCaml) during the first several weeks to exp
  setting. The new material does not include Python.
- To give you an idea of how the course content breaks down by pr
  used Java for approximately 24 lectures and then moved to Pyth
  uses OCaml for approximately 18 lectures and then moves to Jav
- Both versions prepare students for the rest of the CIS curriculum
- The prerequisites remain the same. The difficulty level between th
  standards will be similar.
- We believe that the new material for CIS120 (combined with the r
  stronger Java programmers than the old material, by starting stuc

Harshal S Chhaya
@hschhaya

My son's intro CS class at Penn a few years back was in OCaml.

I was really glad.

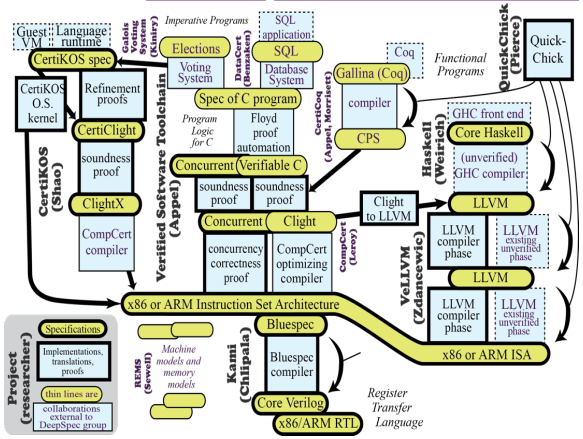
He learned a whole new way to think. And I learned a lot too.

6:36 PM · Jan 10, 2024 · **163** Views

1 4

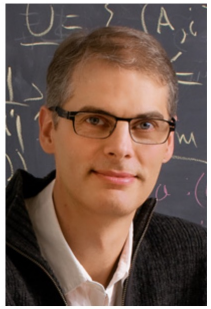
Science of Deep Specification (2017)

deep spec



[Appel](#)
Princeton

[Chlipala](#)
MIT



[Pierce](#)
U. Penn

[Shao](#)
Yale

[Weirich](#)
U. Penn

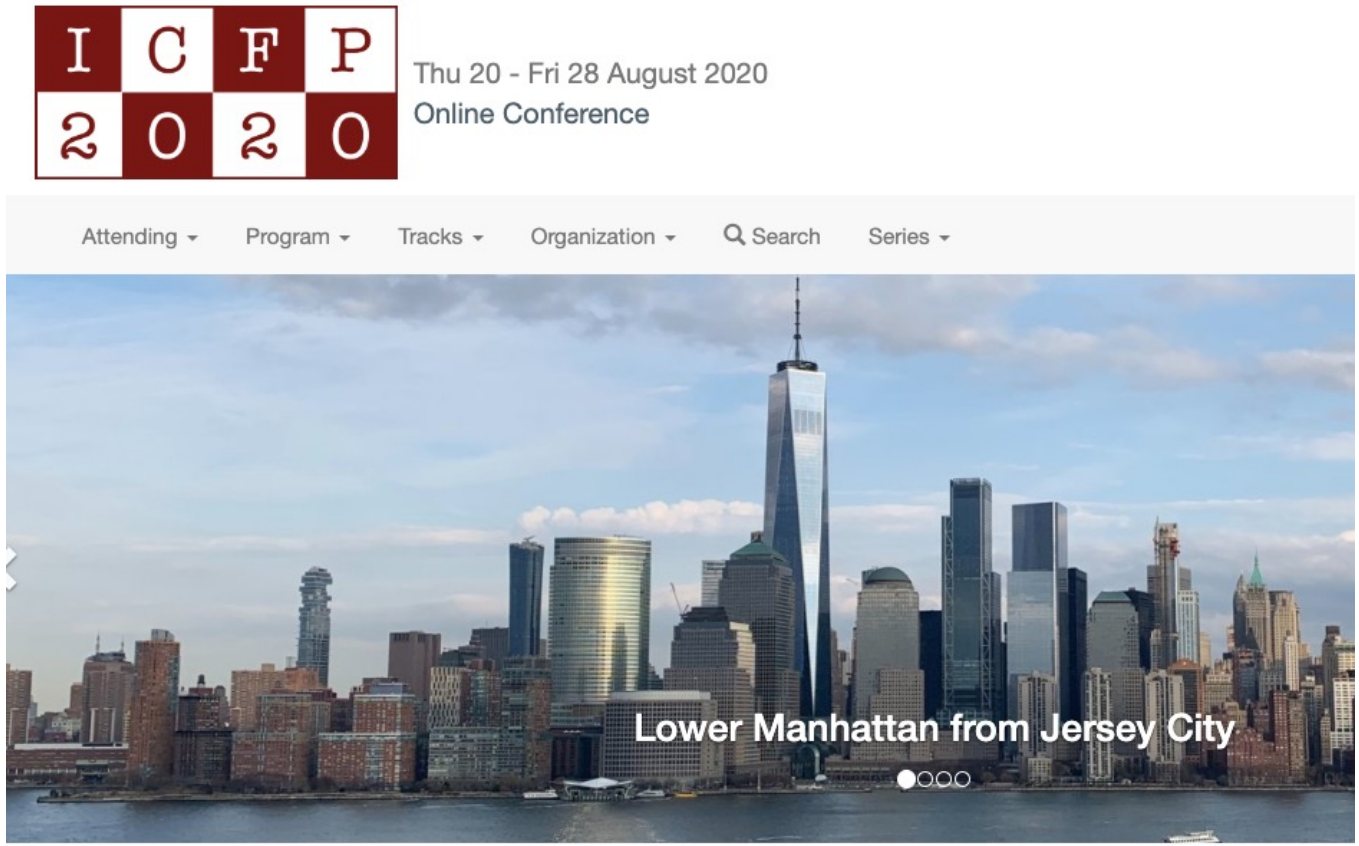
[Zdancewic](#)
U. Penn

[Beringer](#)
Princeton

PL Club:

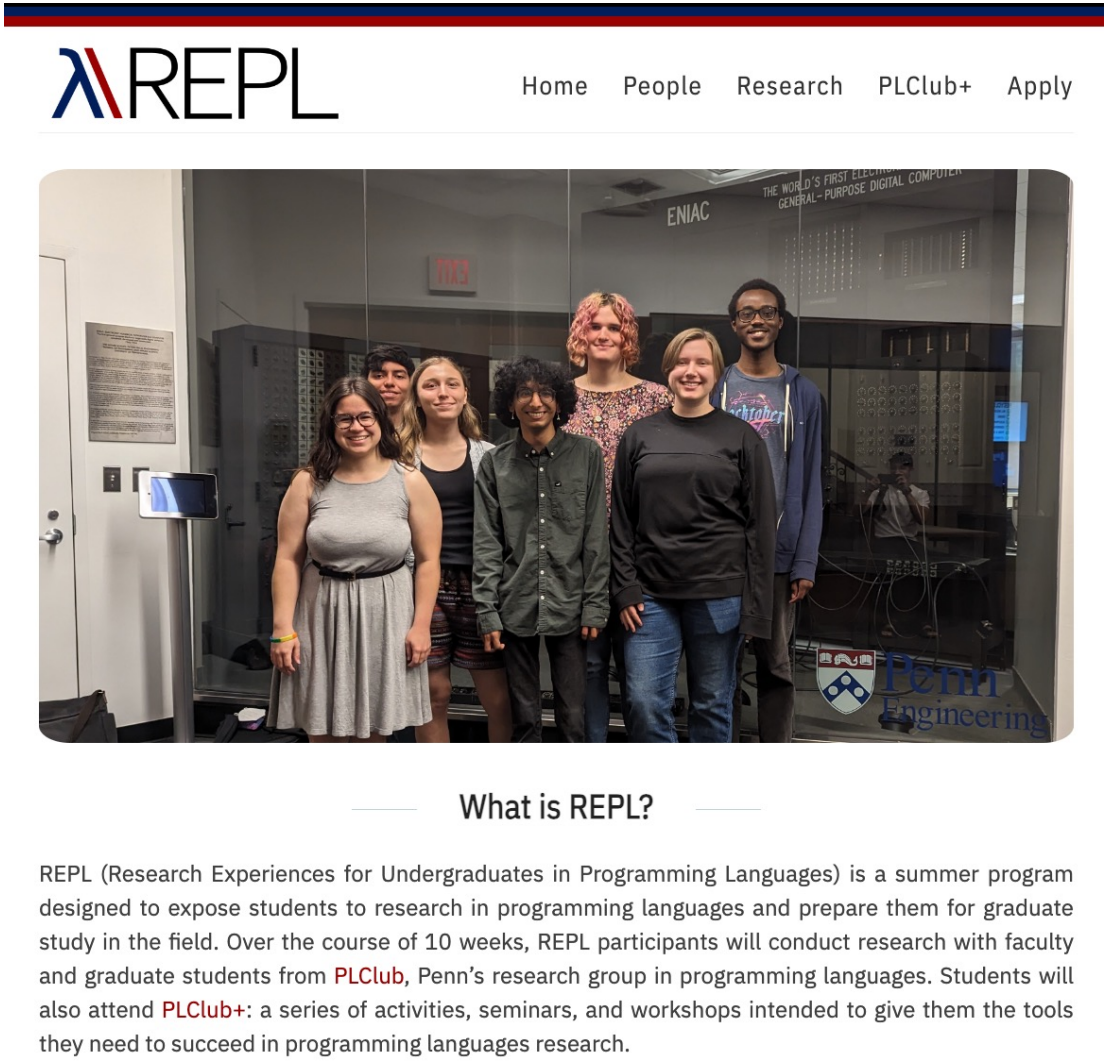
- QuickChick
- hs2coq
- Vellvm
- interaction trees
- verified web server

Virtual ICFP 2020



Research Experiences for undergraduates in Programming Languages (2023-?)

- NSF supported multi-year REU program
- Driving force Ph.D. student: Joey Velez-Ginorio
- PL Club's emphasis on community



What makes PL Club Special?

- Supportive, friendly, inquisitive environment
- High standards of academic rigor
- Ambitious impact
- Fun!

What makes it all work?

Respect everyone

- Your students work **with** you not **for** you
- Your job is to guide them to success
- Their success makes **you** look good

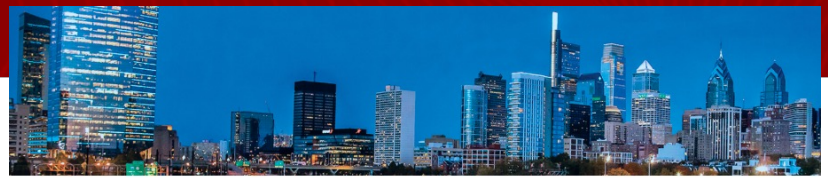
- Recruit students that are smarter than you
- Demonstrate respect: listen to their ideas

Collaborate widely

- I couldn’t have PL Club without Steve and Benjamin
- Students need someone else (besides you) to talk to
- Students need someone else (besides you) to listen to/learn from
- Look for collaborators even if across locations and areas

Develop Community

- Develop shared interests and vision
- Get excited about new ideas together!



NEW JERSEY
PROGRAMMING LANGUAGES AND SYSTEMS
SEMINAR

May 19, 2023
University of Pennsylvania

- Get everyone together at least once a week
- Engage with the group
- Ask students to contribute to the community so that they will value it (i.e. organize seminars, host visitors, give talks, volunteer at conferences)

Build Trust

- Research requires risk. If you knew what you were doing, it wouldn't be research
- **Students** need to trust **you** enough to bring their problems to you (i.e. research means failing fast)
- **You** need to trust **your students** enough to give good notes

Support

- Happy people do their best work

- Check in on your student's non-academic needs
- Give them time and space to solve problems
- Encourage social connections (board games, etc.)

Successful Research Group

- Respect
- Collaboration
- Community
- Trust
- Support

None of this is difficult
But you have to think about it
And it is worth doing

Penn Engineering
University of Pennsylvania

On Building a Successful Research Group

Stephanie Weirich
University of Pennsylvania