Joe Gibbs Politz

joe@cs.brown.edu
jpolitz.github.io
+1 (774) 262-5198

Academic Experience

2015-Present Swarthmore College
Visiting Instructor

2009–2016 Brown University
Expected PhD Candidate, Advised by Shriram Krishnamurthi

2005–2009 Worcester Polytechnic Institute
Bachelor of Science with High Distinction, Major in Computer Science, Minor in Mathematics

Teaching

Courses

2015 **CS91 Introduction to Programming Languages** (cs.swarthmore.edu/~jpolitz/cs91/s15/)
Swarthmore College | Visiting Instructor

CS35 Data Structures and Algorithms
Swarthmore College | Visiting Instructor

2014 **CS1729 Programming Languages Lab**
Brown University | Instructor

2013 **CS1730 Introduction to Programming Languages** (cs.brown.edu/courses/cs173/2013/)
Brown University | Graduate TA | Instructor: Shriram Krishnamurthi

CS1729 Programming Languages Lab
Brown University | Instructor

2012 **CS1730 Introduction to Programming Languages** (cs.brown.edu/courses/cs173/2012/)
Brown University | Co-Instructor | Instructor: Shriram Krishnamurthi

Included an online component that had hundreds of world-wide participants (cs.brown.edu/courses/cs173/2012/).

For a sample of my lecturing style, see www.youtube.com/watch?v=b1MvKYHCdno.

Co-Advised Students

Junsong Li (M.S. 2015) – *Python Semantics, JavaScript Semantics*
Daniel Patterson, (B.S. 2013) – *Python Semantics, Pyret and CaptainTeach*
Sumner Warren, Matthew Milano (B.S. 2013) – *Python Semantics*
Matt Carroll (M.S. 2012) – *JavaScript Semantics and Belay*
Spirodon Eliopoulos (M.S. 2011) – *JavaScript Security Verification*

Research

Pyret (www.pyret.org, code.pyret.org)

I currently lead the development of the Pyret programming language, which is designed explicitly to be a good choice for programming education. This has led to a number of challenging design and engineering choices for the language: a zero-install experience by hosting the language's compiler and runtime within a web browser, a focus on frictionless unit and property testing, and a syntax and semantic design that is familiar to instructors and avoids unpleasant surprises.

In-flow Peer Review (CaptainTeach)

I am currently studying in-flow peer code review, a variant of student code review where students review one another's work at intermediate assignment stages. In two courses at Brown in fall 2013, we built assignments where students first submitted test cases for a function, then reviewed 2-3 other students' test cases, then implemented the function itself. This reinforced test-first development, reading and evaluating code, and technical writing skills.

To support this, we built a web-based learning environment called CaptainTeach, and have published our results and a description of the tool at ITiCSE and ICER 2014. I also helped lead a working group at ITiCSE exploring designs and methodologies for in-flow peer review with an international group of education researchers. The report for the working group is under preparation.

Semantics Engineering for Python and JavaScript

Existing languages in wide use are not necessarily well-understood by the programming languages research community. Some have earned reputations for having abnormal or even pernicious behavior. We take two popular scripting languages—Python and JavaScript—as found objects and build formal models of them by probing their implementations with tests and studying written documentation and (when they exist) specifications. These formal models, in the form of operational, small-step semantics, have their most basic use in providing an account of these languages' behavior in the mathematical domain of programming languages research. Beyond their explanatory power, they also enable more reliable tools for these languages, like type checkers and program analyses, based on existing programming languages techniques.

Type-based JavaScript Verification

JavaScript's ubiquity, combined with its paucity of abstractions, has made the language a target for attacks that exploit language-based security vulnerabilities. Various techniques for *web sandboxing* have been proposed to allow running untrusted JavaScript code safely within a trusted environment (as in the case of a Web page loading a third-party advertisement). In 2011, I designed a novel type-checking strategy for verifying JavaScript sandboxes and applied it to find vulnerabilities in, and eventually give a proof of correctness for, an existing sandboxing library. The type-based technique is general, and a variant of it has also been used to verify privacy properties of Firefox extensions implemented in JavaScript.

Belay & Macaroons

At Google, I worked with a team that investigated protocols and user interface designs for fine-grained, web-based sharing. This included a Google Docs-like interface for storing accounts across the web, and a framework for enabling window-to-window secure sharing between sites within a browser, without requiring server-to-server interaction. The eventual published output of the project was an NDSS paper on *macaroons*, a flexible cloud authorization protocol.

Professional Programming

2012, 2011 **Google** | Intern
Mountain View, California, USA

Developed Belay, Macaroons, and related tools and protocols for secure web-based sharing.

2010 **Fujitsu** | Intern
Sunnyvale, California

Developed tools for statically analyzing JavaScript programs.

2009 **ASSISTments** (www.assistments.org) | Lab Manager/Lead Developer
Worcester Polytechnic Institute, Massachusetts

Managed the continued development of the existing ASSISTments project, a web-based tutor for middle-school math curricula. Managed undergraduate and graduate student contributions and maintained server infrastructure to serve the application to thousands of students.

Books

Programming and Programming Languages, 2015 Edition. Shriram Krishnamurthi and Joe Gibbs Politz. http://papl.cs.brown.edu/2015/

Journal Publications

Type-based verification of Web sandboxes. **Joe Gibbs Politz**, Arjun Guha, and Shriram Krishnamurthi. Journal of Computer Security (JCS), 22(4), 2014.

Conference & Workshop Publications

Slimming Languages by Reducing Sugar: A Case for Semantics-Altering Transformations. Junsong Li, Justin Pombrio, **Joe Gibbs Politz**, and Shriram Krishnamurthi. SPLASH/Onward! 2015.

In-Flow Peer Review. Dave Clarke, Tony Clear, Kathi Fisler, Matthias Hauswirth, Shriram Krishnamurthi, **Joe Gibbs Politz**, Ville Tirronen, Tobias Wrigstad. Working Group Report, Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2014.

In-Flow Peer-Review of Tests in Test-First Programming. **Joe Gibbs Politz**, Shriram Krishnamurthi, and Kathi Fisler. International Computing Education Research Conference (ICER), 2014.

CaptainTeach: Multi-Stage, In-Flow Peer Review for Programming Assignments. **Joe Gibbs Politz**, Daniel Patterson, Kathi Fisler, and Shriram Krishnamurthi. Innovation and Technology in Computer Science Education (ITiCSE), 2014.

Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. Arnar Birgisson, **Joe Gibbs Politz**, Úlfar Erlingsson, Ankur Taly, Michael Vrable, and Mark Lentczner. Network and Distributed System Security Symposium (NDSS), 2014.

Python: The Full Monty: A Tested Semantics for the Python Programming Language. **Joe Gibbs Politz**, Alejandro Martinez, Matthew Milano, Sumner Warren, Daniel Patterson, Junsong Li, Anand Chitipothu, and Shriram Krishnamurthi. Object-Oriented Programming, Systems, Languages & Applications (OOPSLA), 2013.

TeJaS: Retrofitting Type Systems for JavaScript. Benjamin S. Lerner, **Joe Gibbs Politz**, Arjun Guha, and Shriram Krishnamurthi. Dynamic Languages Symposium (DLS), 2013.

A Tested Semantics for Getters, Setters, and Eval in JavaScript. **Joe Gibbs Politz**, Matthew J. Carroll, Benjamin S. Lerner, Justin Pombrio, and Shriram Krishnamurthi. Dynamic Languages Symposium (DLS), 2012.

Progressive Types. **Joe Gibbs Politz**, Hannah Quay-de la Vallee, and Shriram Krishnamurthi. SPLASH/Onward!, 2012.

Semantics and Types for Objects with First-Class Member Names. **Joe Gibbs Politz**, Arjun Guha, and Shriram Krishnamurthi. Foundations of Object-Oriented Languages (FOOL), 2012.

ADsafety: Type-based Verification of JavaScript Sandboxing. **Joe Gibbs Politz**, Spiridon Aristides Eliopoulos, Arjun Guha, and Shriram Krishnamurthi. USENIX Security Symposium, 2011.

Scalability and Robustness in the Domain of Web Based Tutoring. Jozsef Patvarczki, **Joe Gibbs Politz**, and Neil Heffernan. Scalability Issues in Artificial Intelligence in Education (Workshop), 2009.

Activities & Service

2016 **European Conference on Object-Oriented Programming |** External Review Committee

2015 **Dynamic Languages Symposium (DLS) |** Program Committee

2011-2015 **The Brown PLT Blog** (blog.brownplt.org) | Author, Designer

2009-2014 **Brown Computer Science |** Bytesoccer (Intramural League)

2014 **Vail Computer Elements Workshop** (vcew.org) | Session Chair: Security Research

2014 **ITiCSE Working Group on In-Flow Peer Review |** Co-organizer

2014 **Workshop on Dynamic Languages and Applications (DYLA) |** Program Committee

2012-2013 **Brown Computer Science |** Faculty-Graduate Liaison
From the Brown web page: "The FGL is a senior PhD candidate tasked with handling most faculty-grad interactions and concerns... Responsibilities include monitoring [grad student volunteer] activity, allocating office space, and controlling access to the grad nest egg."

2013 **Brown Computer Science |** PhD Admissions Committee

April 2012 **Hack the Future** (hackthefuture.org) | Volunteer
August 2011 At two one-day Saturday workshops, taught middle-school students basic programming techniques in Unity 3D (2011) and in WeScheme (2012).

2011 **Brown Computer Science |** Hospitality Czar
From the Brown web page: "Hospitality Czars organize the graduate-student recruitment weekends. This includes setting up faculty/student talks, organizing lunches and tours, and arranging housing."

Talks

Bolding *indicates invited talks.*

“Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud.” Georgia Tech Security Seminar, March 2015.

“Python, the Full Monty.” Dagstuhl Symposium on Analysis and Verification of Scripting Languages and Frameworks, 2014.

“Belay: Distributed Authorization in the Cloud.” Vail Computer Elements Workshop (VCEW), 2013.

“Python, the Full Monty.” SPLASH/OOPSLA, 2013.

“A Tested Semantics for Getters, Setters, and Eval in JavaScript.” Dynamic Languages Symposium, 2012.

“Progressive Types.” SPLASH/Onward!, 2012

“A Tested Semantics for Getters, Setters, and Eval in JavaScript.” Dagstuhl Symposium on Foundations for Scripting Languages, 2012.

“ADsafety: Type-based Verification of JavaScript Sandboxing.” Dagstuhl Symposium on Web Application Security, 2012.

Awards

Brown University Sigma Xi Award for Outstanding Graduate Research in Computer Science, 2015
WPI Salisbury Prize, 2009
WPI Computer Science Outstanding Junior Award, 2008