Abstract
Information leaks are more and more prevalent. This should be no surprise: not only are users sharing more information, but there is a growing amount of code that handles sensitive data. Unless we make it easier to create secure programs, information leaks will only get worse.

I present a language, Jeeves, where programs preserve privacy and security properties by construction. In Jeeves, the programmer specifies expressive information flow policies separately from other functionality and relies on the language runtime to automatically differentiate behavior. Jeeves is the first language to factor out information flow, preventing information leaks by reducing the opportunity for programmer error. To help programmers reason about Jeeves programs, I defined what it means for Jeeves to enforce programmer-specified policies and proved that Jeeves correctly does so. To demonstrate the feasibility of this programming model in practice, I implemented Jeeves in Scala and Python and created Jacqueline, a Jeeves-based web framework that enforces policies end-to-end. I describe how I used Jacqueline to implement a course manager, a health record manager, and a conference management system that has been used for an academic workshop.

Biography
Jean Yang received an AB from Harvard University and is a PhD candidate in MIT’s Computer Science and Artificial Intelligence Laboratory (CSAIL). Jean’s research interests are in programming language design and software verification. Jean is a recipient of the National Science Foundation Graduate Research Fellowship, the Facebook Fellowship, and the Levine Fellowship. Jean’s work on Verve, an operating system automatically verified end-to-end for type safety, won Best Paper Award at PLDI 2010.