Teaching Statement

Philipp Haller

I have always enjoyed teaching, as a teaching assistant at EPFL, as a lecturer at summer schools for graduate students, and as a trainer for Typesafe organizing workshops for professional software engineers.

I gained most of my teaching experience as a teaching assistant (TA) at the programming methods laboratory (LAMP) at EPFL. I am particularly proud of my role as TA for the Fall 2009 course Foundations of Software– as the sole TA I designed a new student project based on the Coq proof assistant, the first time this tool was used in the course. In 2008 I received an EPFL Prize of Excellence for an exceptional teaching contribution by re-designing the entire exercise portion of our Advanced Programming course, changing the format of the exercises to self-contained mini projects; in addition, I organized new activities for the lab sessions, such as code walk-throughs, which were received very positively. Over the course of four years I was TA for 6 courses: Compiler Construction (Fall 2006, Fall 2007), Advanced Programming (Spring 2007, Fall 2007), and a graduate-level course on the theory of programming languages and type systems, called Foundations of Software (Fall 2008, Fall 2009). In all of these courses I helped design the weekly or bi-weekly exercises and exam problems. In several of these courses I was also the head TA which included shared administrative responsibilities, fielding of all student inquiries, and oversight of student teaching assistants.

In the past 18 months I was fortunate to gain substantial experience with new massive open online course (MOOC) formats. Working closely with the Scala team at EPFL, I assisted with their MOOC on Functional Programming Principles in Scala, in particular concerning the detection of plagiarism and the empirical course evaluation. This immensely successful course with more than 100'000 registered students to date, boasting one of the highest completion rates (19.2%) for its size, has explored new ways to interactively provide feedback to students while they are working on their homeworks, through automated submission and testing in the cloud. I have co-authored an experience report [1] on this MOOC evaluating surveys of more than 12'000 respondents world-wide on a variety of questions on their experience learning functional programming principles. At Typesafe, I have directed a new commercial offering for personalized tutorial sessions for the Coursera MOOCs "Functional Programming Principles in Scala" and "Principles of Reactive Programming". This new program became quite popular, especially for the course on reactive programming, showing the potential for such new forms of interaction in the context of MOOCs.

Summer schools for graduate students have always been a very inspiring experience for me, both as a participant and as a lecturer. In 2011 I was an invited lecturer at the UPMARC Multicore Computing Summer School. For my two days of lectures, I prepared a set of course materials on a variety of topics on multicore programming in Scala. In most of the lectures, I presented concepts that I had been working on as a Ph.D. student or Postdoc, such as actors, uniqueness types, and applications to large-scale, distributed machine learning. The lectures were accompanied by a web site, which is still visited frequently, with the course materials, suggestions for further reading, and other resources online. At the International Summer School on Trends in Concurrency in 2008, I helped design and teach lectures for graduate students on concurrent programming in Scala using actors and join-calculus-style synchronization based on my graduate work.

Apart from my teaching experience in the classroom I have extensive experience with trainings and workshops for professional software developers. At Typesafe I have created new training materials for a two-day course on concurrent programming with Scala and the Akka event-driven middleware. I have taught this course as well as courses on introductory and advanced Scala programming repeatedly in public trainings as well as in-house trainings for both small and large private companies in the USA, UK, France, Germany, Switzerland, and Hungary.

Additionally, I have been fortunate to work alongside of a number of undergraduate and young graduate students at EPFL as a mentor and supervisor. My involvement ranged from supervising summer interns that helped on various projects around the Scala compiler and libraries, to regular semester and Master's thesis projects, most of which surrounded my Ph.D. work on Scala Actors. I was also one of the supervisors of an undergraduate research project that led to a publication [2] in the proceedings of the 25th International Workshop on Languages and Compilers for Parallel Computing (LCPC 2012). In my mentoring, I focus on solving problems that are most important for a particular project. Through discussions that illustrate how to effectively communicate about problems and possible solutions I strive to give students confidence that they can solve problems independently and communicate effectively about them.

In my role as a new faculty member I would be excited to teach undergraduate and graduate courses on functional programming principles, software foundations (focusing on type systems and formalization), compiler design, and concurrent programming. When teaching, I strive to illustrate concepts and techniques using real-life examples whenever possible. From experience I know that devising good examples is hard work, yet worth the effort, and often makes teaching itself much more rewarding for both instructor and student. Furthermore, I believe in learning by doing, and prefer to give students plenty of opportunities to "get their hands dirty." My involvement with the development of the Scala programming language and Typesafe's Akka open-source project would provide exciting opportunities for students to work with (and perhaps even contribute to) real, open-source programming systems and frameworks.

Last but not least, working together with students is a great opportunity to spread the excitement and intrigue afforded by research, and to inspire them to pursue their own research projects early on in their careers.

References

[1] H. Miller, **P. Haller**, L. Rytz, and M. Odersky. "Functional Programming For All! Scaling a MOOC for Students and Professionals Alike". In: *ICSE SEET*. 2014.

[2] A. Prokopec, H. Miller, T. Schlatter, **P. Haller**, and M. Odersky. "FlowPools: A Lock-Free Deterministic Concurrent Dataflow Abstraction". In: *LCPC*. 2012.