Derek G. Murray — Teaching Statement

Teaching is a fundamental component of an academic career, and the opportunity to teach is one of the driving factors in my pursuit of a faculty position.

Teaching interests My teaching interests extend across three main areas of the curriculum:

The **introductory subjects** expose students to computational thinking and cover the basic building blocks of computer science, such as entry-level programming, computer architecture, digital electronics, and discrete mathematics. Since they are fundamental, these courses should be taught by experienced faculty, but I aspire to teach at this level later in my career.

The **core subjects** provide the student with both the skills that all computer scientists should possess, and more importantly an understanding of what happens inside a computer. In this area, I would be prepared to teach classes on operating systems, networking, parallel programming, distributed systems, and databases.

Finally, the **advanced topics** are geared towards students who are keen to specialize in a particular area, and may have an interest in continuing on to research in that area. Accordingly, I would teach classes that align more closely with my own research interests, such as data center computing, advanced operating systems, parallel algorithms, and distributed databases.

Experience At the University of Cambridge, it is unusual for graduate students to lecture undergraduates directly, or serve as teaching assistants. I therefore sought out other opportunities to teach.

The principal form of graduate teaching at Cambridge is undergraduate supervision, in which small groups of three (or fewer) students meet with a supervisor on a (bi-)weekly basis. A typical supervision includes discussing answers to a take-home exercise, clarifying students' uncertainties with the course material, and ideally delving into aspects of the material that were not covered in the lecture component. I have supervised groups in the following courses: (first year) operating systems, computer design, digital electronics; (second year) distributed systems, C & C++; (third year) advanced systems topics, information retrieval.

My most significant lecturing experience was at the 2011 ISSNet Summer School in Calgary, Canada. At the summer school, I taught a whole-day course on data center computing, including a lecture-based component and an extended practical exercise. The practical exercise gave each student the opportunity to write distributed programs (using the CIEL system that I developed during my Ph.D) on a cluster of virtual machines leased from Amazon EC2. The feedback for this course was positive (average 4.0 rating out of 5), and the practical exercise was adapted for the R202 Data Centric Networking graduate-level course at Cambridge. In addition, I have given invited lectures on this topic to graduate students at Cambridge and various summer schools.

In addition to taught courses, project work is a vital component of a degree in computer science. Moreover, the ability to supervise graduate students is one of the principal attractions of an academic career. At Cambridge I supervised four projects: two development-based final-year undergraduate projects, and two research-based Masters projects. Each project considered a different aspect of parallel programming, and I was impressed by what many of the students were able to achieve in a short time.

Contemporary issues The most pressing issue in computer science education is increasing engagement with underrepresented groups. High-school–level courses are overly focused on the use of particular tools and techniques, and offer poor preparation for the analytical component of a computer science degree since they can create the mistaken impression that computer science and "programming" are equivalent. As an undergraduate, I participated in outreach programs that introduced the relevance of computational thinking to school students, and I would welcome the opportunity to continue this outreach both within and outside the university.

Massive Open Online Courses (MOOCs) offer one opportunity for outreach, and could allow potential students in computer science to try out a course before committing to studying it. While I believe that a student's physical presence in class is valuable both socially and intellectually, I would be keen to develop MOOC versions of my courses, because opening lectures to a wider audience invites broader feedback, which would serve to improve the course in future.

Finally, computer science plays an important supporting role in other disciplines, and up-to-date research in distributed computation can be very relevant to people in disciplines such as physics, applied mathematics, statistics, and experimental social science. My own experience of interdisciplinary courses—as a Masters student in a physics department—was that modern systems research has had little influence on how scientific computing is taught, and the current tools lag behind the state of the art by decades. Recent years have shown how research into large-scale distributed computing can transform information-based industries, and teaching these techniques more widely offers the chance to transform other disciplines in the same way.