2019 **Tertiary Teaching Excellence Awards**
General Category

Associate Professor Andrew Luxton-Reilly

School of Computer Science
The University of Auckland

Contents

Whoever is fundamentally a teacher takes all things seriously only in relation to his students, including even himself.

Friedrich Nietzsche

Excellent Teaching in Computing Education

For me, excellent teaching is about sharing my passion for computing, stimulating curiosity and conveying to students why Computer Science *matters* through its enormous capacity to improve the world. Excellent teaching should help students learn effectively, be self-reflective about both content knowledge and professional behaviour, be aware of the connections between the digital world and the social world, and understand how choices in one world affect the other. Moreover, they should have positive experiences as they acquire these characteristics. My students report that they enjoy being challenged, asked to think, and to solve problems in class; and that they value being part of a supportive, respectful learning community. Excellent teaching goes hand-in-hand with enjoyable learning.

Although Computer Science (CS) has strong mathematical and logical foundations, the digital world and the social world are intimately connected. I believe computing educators should take a holistic view of the discipline and ensure students are connected to people and society. It is essential for CS students, who develop technology that has a profound effect on our future, to be able to communicate effectively with people and to have an appreciation of ethics and social justice.

Two main principles underpin my entire teaching practice. First, I design learning activities that use logical steps to scaffold the acquisition of knowledge, emphasise connections between different branches of knowledge, and encourage reflection on the broader implications of that knowledge. The second principle is that teaching practice must be based on a continuous cycle of research-informed, critical reflection.

There are challenges unique to CS that effective teaching practices and learning design must address. The discipline is highly technical and requires a great deal of precision, as a single character out of place in a program will typically render a solution unintelligible to a computer. These same programs must also be designed to be easily understood by other people so that the software can be maintained. Rapid changes in technology require students to develop a wide range of both technical and professional competencies, and work at multiple levels of abstraction. In addition, students who study CS have wildly differing levels of prior experience, and the discipline is plagued by negative stereotypes and implicit bias.

These factors result in a discipline that has one of the lowest levels of student engagement as measured by national student surveys, and among the worst gender equity of any subject (typically less than 20% of CS students are women). [1] Developing a more inclusive and

[1] Morgan, Sinclair, Butler, Thota, Fraser, Cross, and Jackova. (2018). Understanding International Benchmarks on Student Engagement: Awareness and Research Alignment from a Computer Science Perspective. In *Proceedings of the 2017 ITiCSE Conference on Working Group Reports* ACM, NY, USA, 1-24.

effective learning environment is one of computing's grand challenges.[2] These same challenges make it one of the most interesting subjects to teach.

Career Overview

Over my 24-year career, I have taught a variety of class sizes (from three to 850 students), and learners (high school students, adults engaged in continuing education, undergraduates and graduates) and have supervised more than 60 postgraduate students (ten PhD, seven Masters, 50 Honours).

My career has had an unusual trajectory. At university I chose an eclectic range of undergraduate subjects to broaden my outlook but had poor study habits, so I failed several courses. I empathise with struggling students and teach with compassion because of these early experiences. Through my BA and MA (Hons) in Philosophy, I acquired an interest in contributing positively to society and learned to think broadly about the history and influence of *ideas* on society. I learned to be critical of both sides of an argument; this has been invaluable in developing a student-centric view of teaching and a reflective approach to practice. In parallel to my studies in Philosophy, I completed a BSc in Computer Science to satisfy my interest in cognitive systems (mind, language and logic).

I began in the Computer Science Department (now the School of Computer Science – SoCS) as a teaching assistant in 1994 while completing my MA, and started teaching undergraduate courses in 1995 as a Tutor, then later as a Senior Tutor. For the first decade of my career I focused exclusively on teaching, receiving a Teaching Excellence Award from the School of Mathematical and Information Sciences (1996), and a University of Auckland Teaching Excellence Award (2003).

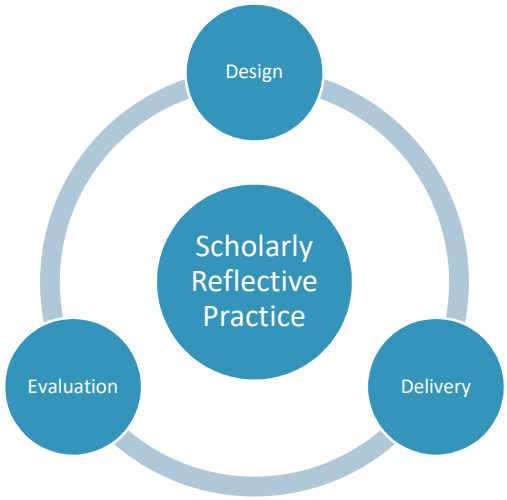
[2]McGettrick, Boyle, Ibbett, Lloyd, Lovegrove and Mander (2005) Grand Challenges in Computing: Education – A Summary, *The Computer Journal*, vol. 48, 1, pp. 4248.

In 2005, I became frustrated with my limited knowledge about the most effective teaching strategies for CS content. I adopted a more critical and scholarly approach to teaching, seeking rigorous evidence for decisions that impacted practice. This led to a PhD (2008–2012) investigating learning through student-generated questions in CS, and a shift to a more traditional research and teaching role as a Senior Lecturer.

In the Classroom

Since my research area is CS Education, the classroom is the core of all my professional activities and is truly the nexus between teaching and research.

My PhD followed an Education Design-Based Research paradigm.[3] In line with this I have adapted my teaching using established theory and evidence-based practice. I use this approach to design learning experiences, deliver those experiences, evaluate the effectiveness, and use the evaluation to feed back into the design for the next iteration.



My teaching practice contextualises computing. I get students to understand computing as multi-faceted and socially important, arousing their curiosity, and demonstrating processes that help them build skills and knowledge through independent and collaborative activities.

Being guided and taught by Andrew has made me believe that I can use my education to make a real difference in society.

Nazish Khan, former student, 2019

I aim to ensure my students are intrinsically motivated. They learn best when they are actively seeking knowledge rather than 'jumping through hoops'. Student curiosity can be developed by making connections between components of knowledge, and explicitly demonstrating *processes* (e.g. developing and testing hypotheses) that lead to successful knowledge building. Such processes are enhanced through interaction with other students.

In this modern age when there's a temptation to just listen to the lecture recordings online and read the slides in our own time, I found Andrew's lectures to be invaluable to attend in person, as the level of engagement was second-to-none.

COMPSCI 107 student, 2018

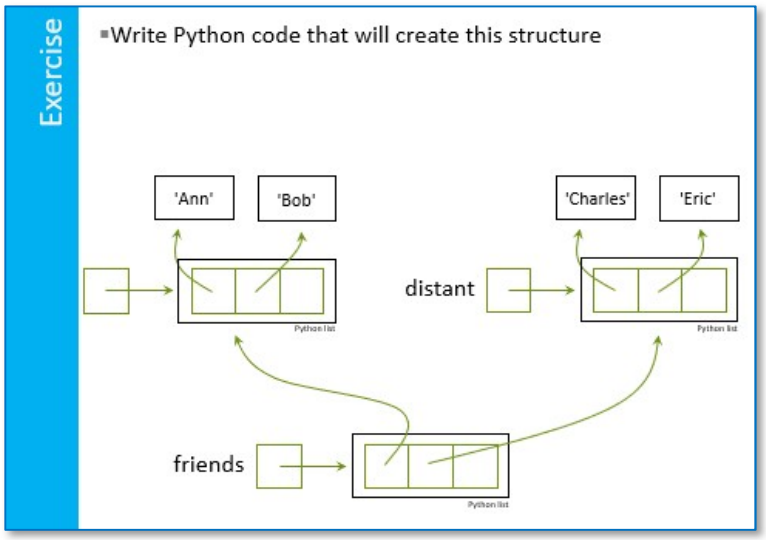
---

[3] Luxton-Reilly, A., (2012) The design and evaluation of StudySieve: A tool that supports student-generated free-response questions, answers and evaluations. PhD Thesis. University of Auckland.

Making Connections and Applying Knowledge

CS requires precision, attention to detail, and the understanding of highly interdependent concepts. To reduce extraneous cognitive load the exposition of concepts must be carefully structured and the relationships between them made explicit.

Material is introduced and reinforced in lectures through a variety of activities involving peer learning. This promotes engagement, improves social interaction and communication skills, and gives students the opportunity for immediate feedback.



Andrew … built up our knowledge step-by-step. Lectures … were active … showing us how data structures work, why the advantages and disadvantages of a certain data structure are what they are. Andrew often posed questions to challenge our understanding (I usually found myself thinking 'good question') …. many of my classmates would volunteer their thoughts about how to make some program more efficient or mention alternatives that they felt would work better. These remarks were always encouraged.

Jo-Anne Coxon, COMPSCI 107 student, 2018

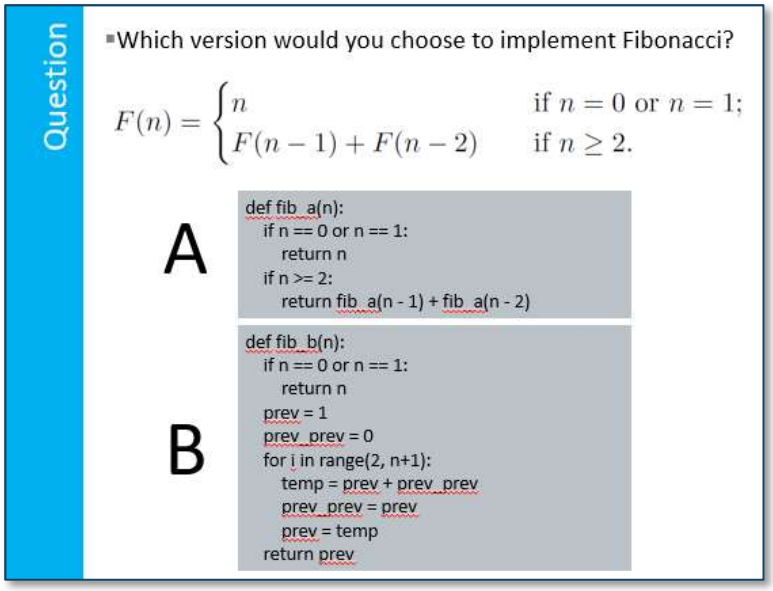
In each lecture I ensure there are several exercises distributed throughout the content. After introducing new concepts or skills, I ask students to demonstrate their understanding by completing an activity or answering a question. Students are asked to work alone, then discuss their answers with their peers before contributing them to a class discussion.

His application/method of approaching a concept in the initial stages and then expanding upon its analysis and depth was amazing.

COMPSCI 101 student, 2014

Arousing Curiosity

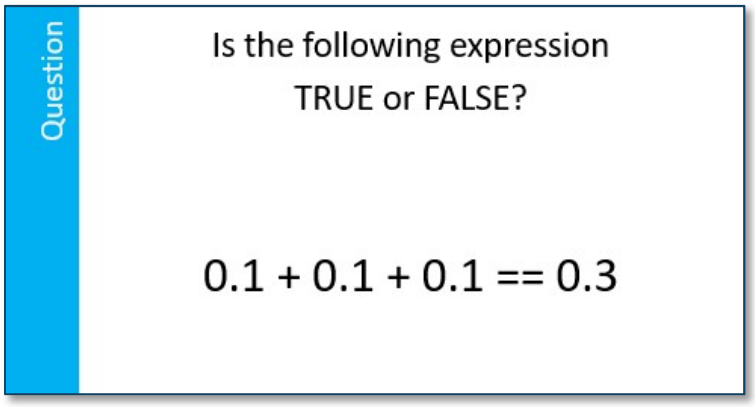
Cognitive dissonance can be a powerful tool to motivate curiosity. For example, I ask students to estimate the time each program in the example on the right takes to calculate the 100th Fibonacci number. Students typically estimate 1-2 seconds. When I reveal that one version takes less than 1 millisecond while the other takes over 800,000 years, an audible gasp is heard, followed by a buzz of conversation. This motivates a discussion of the potential social impact of choosing an inefficient algorithm, and the global cost of algorithms (e.g. Google queries are estimated to generate 500kg of CO_2 per second), bringing life to a dry theoretical topic.



He […] has a way of getting straight to the point and yet teasing out interesting cases that really make you think.

COMPSCI 107 student, 2018

The intuitive answer in the example below is incorrect due to the limited precision of floating point numbers.



The consequences of limited precision errors are highlighted with historical examples of the social cost of mistakes.

Inclusive Teaching and Learning

I create a safe and inclusive learning environment by setting ground rules that insist on mutually respectful interactions. During introductions I discuss my whanau, my personal story, and how it has impacted the lens through which I see the world.

As a mature student going back into tertiary education for the first time in over fifteen years, I found Andrew to be an excellent communicator of sometimes difficult concepts, and he did so with enthusiasm, charisma and warmth.

COMPSCI 107 student, 2018.

Respect. That two-way street that can but shouldn't be underrated. The majority of students respected Andrew which paved the way for an environment that was attuned to listen, learn and ask questions.

COMPSCI 101 student, 2014

I ensure that I am approachable and listen carefully to students, aiming first to understand where they are coming from, and being aware of any potential subtext. In this way, I can answer questions more effectively, and can offer pastoral care.

Not only did Mr. Luxton provide me the support that I so needed in the subject but he has been a key factor in bolstering my confidence in my own capabilities. He was always there in the times, when without some kind of extraneous educational and mental support, I would have crumbled. […] To me, this man alone deserves the entire credit for the foundation that was laid in precarious times and which now I am robustly building upon. […]

COMPSCI 101 and COMPSCI 105 student, 2001

Over my career I have been Enrolment Advisor for eight years, participated in Girls into Science for four years, International Student Welfare officer for six years, and a member of the Student Support Group for 16 years. These roles have given me insight into some of the difficulties encountered by students, and empathy with their personal situations.

I'm a study abroad student from the U.S.... how you handled the situation tonight really impressed me.... I just thought you deserved a thank you. You've really changed my perspective on what the professors here can be like.... Thanks again and keep it up, you're awesome.

COMPSCI 130 student, 2019

To address the challenge of gender imbalance in CS I ensure students are aware of gender equity issues, such as bias, negative stereotypes, and perceptions of identity. Students are asked to engage in activities such as implicit bias training and write reflective accounts of their own experiences.

Software increasingly influences society, so I try to raise awareness of the social implications of software design early in students' CS education using examples. I ask students to consider the racial bias of image recognition software and binary, male/female gender options that contrast with classification systems that allow for more gender diversity.

Report from NZ Herald Dec 8th 2016

Richard Lee was told his eyes were not open when submitting an image of himself to New Zealand's online passport photo checker.
Photo / Richard Lee Facebook

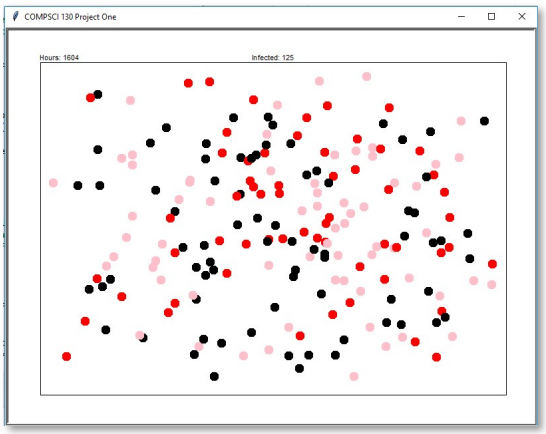
Activity

- Why might the following code be problematic?
  - How might it be improved?

```
gender = input('Enter your gender (M/F): ')
if gender.lower() == 'm':
    #do something appropriate for males
else:
    #do something appropriate for females
```

Glad to see how inclusive the course content is regarding gender, race, etc.
CS705 /SE 702 student, 2017

My previous research (with student Nazish Khan, below left) has investigated the potential benefits of using socially relevant contexts for computing assessments.[4] For example, I have required students to develop a simulation to model the spread of virus among an unvaccinated population.



Teaching and Learning Design

Effective learning design must be informed by what works for students – which activities will arouse curiosity for different students? Where do students have the most difficulty? How can feedback have the greatest impact? What is the student experience in this course? Evidence-based research[5] helps me design course delivery methods that answer these questions.

Constructive Alignment in COMPSCI 130

Feedback from industry, students and academics indicated that students were not able to program as fluently as expected as they progressed through our program and graduated with a CS major. To address this feedback I am currently leading a step-change in the delivery and assessment model used in our core software courses, starting with a first-year programming course, COMPSCI 130.

The delivery model shifted to one introductory lecture and two, two-hour programming laboratories each week. Assessment has moved from paper-based examinations to computer-based practical programming tests. The tests have open access to the internet to ensure an authentic experience that aligns with course learning outcomes, graduate capabilities and industry expectations. Students have reported the course is extremely challenging but find the new approach valuable.

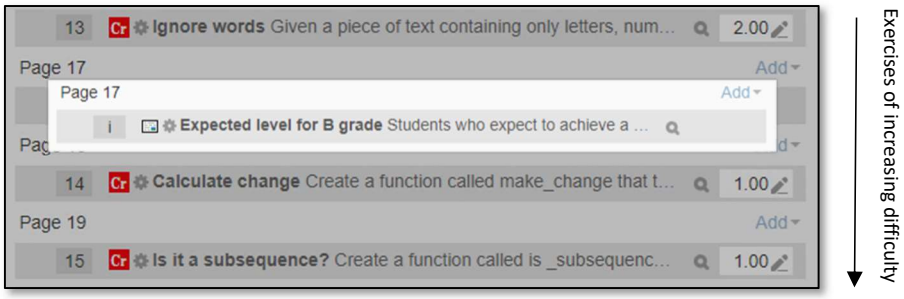
[4] Khan, N. & Luxton-Reilly, A. (2016). Is computing for social good the solution to closing the gender gap in Computer Science? In Proceedings of ACSW '16. ACM, USA

[5] Luxton-Reilly**,** Simon, Albluwi, Becker, Giannakos, Kumar, Ott, Paterson, Scott, Sheard, and Szabo, (2018). Introductory Programming: A Systematic Literature Review. In *Proceedings Companion of ITiCSE'18* ACM, NY, USA, 55-106.

Writing code in a timed environment was a much more valuable experience than the multi-choice and handwritten tests I've done in the past. I hope you continue doing them. Thank you!

COMPSCI 130 student, 2019

To ensure that all students can make progress while receiving clear feedback on their performance, all assessed activities are differentiated by expected performance levels.



Staff have responded positively, and we are currently tracking student performance in subsequent courses to verify the effectiveness of the new approach.

The new format for this course is fantastic. The 4 hours of labs to 1 hour of lectures in a regular semester is an amazing way to teach programming fundamentals.

COMPSCI 130 student, 2019

Developing Professional Ethics in Introductory Courses

COMPSCI 101 focuses on the syntax and semantics of programming languages, but I also wanted students to appreciate the ethical and professional issues arising in programming careers. Initially, I introduced this topic as a week-long module and assessed it with a short essay. Students hated the assignment and disengaged, obtaining only superficial understandings. To address this, I trialled an exercise that had a lower entry barrier, allowed for a range of views, and examined a topical issue, such as Edward Snowden, or accountability on social media. Students were more positive about this exercise, and the approach was reported in the Faculty of Science Newsletter.

Because Computer Science students are primed to concentrate on the computational side of things at university, Andrew makes sure discussions and written tasks are embedded in a context that the students can relate to, giving real purpose to the exercise.

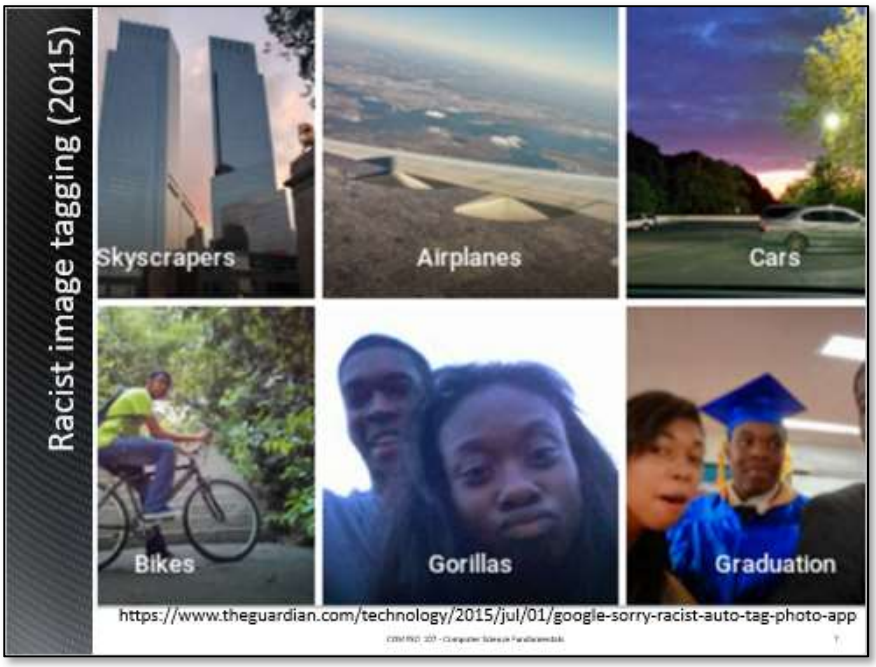
https://www.sciencestaff.auckland.ac.nz/en/about/news/news-2014/2014/08/literacy-and-ethics.html

---

Using the cycle of feedback, reflection and improvement, I have refined the activity. In 2018, students were expected to write 200 words on topics related to computing ethics, professionalism, technology or society for each of ten weekly labs. Embedding writing in this way has resulted in CS students, who typically avoid written tasks, writing at least 2000 words about broader issues. For example, a summary of the first weekly task is:

Complete the Microsoft unconscious bias training, discuss how you think unconscious bias might affect the academic study of Computer Science at the University of Auckland and post a succinct written summary of your views.

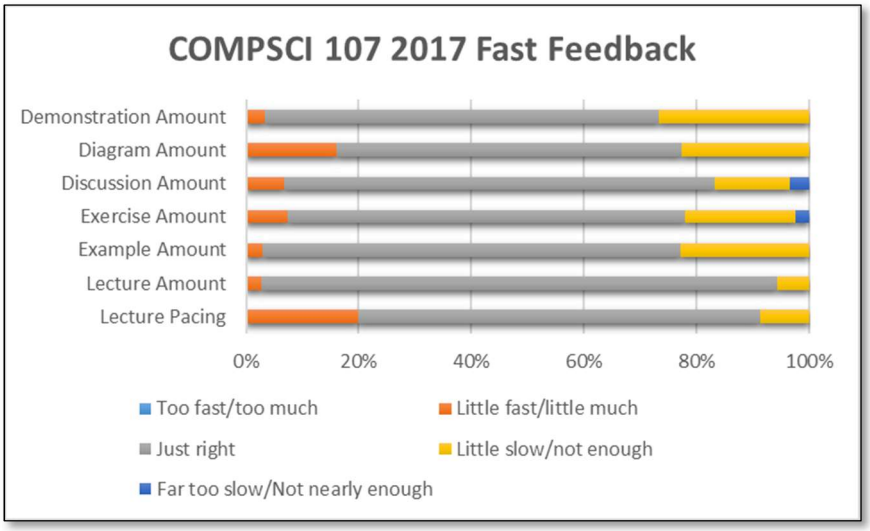
Other topics include the impact of software failure, responsibilities of computer professionals, creating inclusive and usable user interfaces, and the impact of artificial intelligence.



These activities achieve multiple aims: encouraging curiosity, developing written and oral communication skills, exposing students to a varied range of viewpoints, raising awareness of the real impact of technology, and establishing standards of professional behaviour.

Reflection and Feedback to Improve Course Curricula

Feedback from students and colleagues plays a central role in the iterative development of course content and curriculum. For example, I use regular fast feedback questionnaires to initiate discussion with students about learning.



I became course coordinator for COMPSCI 111 in 1997 and iteratively modified the curriculum over the following 10 years. A variety of technologies were trialled using data obtained from observations, student surveys, analysis of assessed work and feedback from tutors.

> *Andrew […] is always receptive to suggestions from both students and colleagues, and he will put those suggestions into effect if at all possible.*
>
> *Ann Cameron, PTF, 2002*

After each course, I reflect on both content and delivery and plan appropriate changes for the following semester. For example, for a service course covering a variety of rapidly changing technologies, I implemented a modular curriculum which had common threads developed across the course, but in which each module could be replaced with a variety of different software tools/technologies.

> *I have had the opportunity to work with Andrew over the past two years and have been impressed with his ability to develop curriculum material in a comprehensive and coherent manner. […] Andrew has really done an outstanding job at curriculum development.*
>
> *Professor Myra Cohen 2002*

In 2005, I leveraged student feedback to redesign the curriculum and course resources to serve both Science students and General Education students. Student satisfaction improved to over 90%, reaching 98% positive responses in 2006. My resources have continued to be used for delivery of COMPSCI 111 since 2005 by more than 15 CS academics.

> *The relaxed (but professional) lecturing style reflected his knowledge, enthusiasm and interest in the session, and was well structured and organised.*
>
> *Associate Professor Dan Exeter - peer review COMPSCI 111 2007*

Innovation in Teaching

I use a variety of technologies to support flexible learning. I began using the world-wide web in 1996, when it was less than two years old (and consisted of 50,000 documents), and have been an early adopter of many, now standard, technologies such as wikis, forums, blogs, and video-recorded lessons.

These technologies enable collaborative learning approaches to be applied in large classes. For example, in the early 2000s I asked students in COMPSCI 111 to collaborate using a wiki to develop a chapter of an online textbook that provided resource material for students in a subsequent year.

The worksheets we got in class were really great.... Studysieve was a valuable resource. Overall, a really great course!

COMPSCI 111 student, 2011

I have been awarded ten University learning and teaching grants for innovative approaches to teaching and learning, including: using videos to demonstrate program development techniques (2005, 2006); developing and evaluating e-learning tools to support collaborative teaching and learning (2008); customising feedback for students in very large classes (2008); improving timeliness of feedback through automated marking (2008, 2010); using online peer assessment effectively (2009); and enhancing tools that support student-generated content (2009, 2010, 2011).

Andrew has … a deep understanding of the pedagogical process and a willingness to experiment and improve on his teaching to the strong benefit of his students.... His teaching methods allow for all modes of learning, and he is one of the earliest and most enthusiastic adopters of flexible learning and web-based resources in the department.

Professor John Hosking, HoD, 2003

Among other innovations, these teaching and learning grants helped support the refinement and evaluation of peer learning systems such PeerWise (led by Paul Denny) and Aropä (led by John Hamer) that are now widely used by an international audience.

In my teaching practice I have experimented with a variety of approaches, including: lecture activity worksheets; live coding demonstrations; role-playing computing functions; large class think-pair-share activities; flipped classrooms; studio-based learning; contributing student pedagogy; pair programming; collaborative tests; and creative open-ended competitions.

Andrew is an innovative teacher, and receptive to many new ideas and techniques informed by his extensive knowledge of the educational literature.... I observe[d] his approach for administering group tests, in which students work collaboratively to solve questions similar to those encountered on a test sat immediately prior and individually. Andrew … had developed novel protocols for assigning students to groups. It would have been much simpler to conduct a test in a more traditional manner, but … Andrew was driven to explore this new approach by a desire to engage students in collaborative problem-solving and promote effective learning.

Dr Paul Denny, 2018

In writing code for the class Andrew made credible mistakes on purpose … we could follow … each step he was taking and [learn] how to solve mistakes we would probably make.... found the content a breeze with this approach.

COMPSCI 101 student, 2014

Leadership in Teaching

I aim to improve the quality of the student experience in CS through embedding high-quality resources within a well-structured curriculum, and by helping other staff realise their teaching potential.

Programme and Curriculum Review

I was a member of the Faculty working group restructuring our BSc and introducing a new Bachelor of Advanced Science (Honours) degree in 2016–2017, and led the development of new majors for several programs offered by the SoCS.

This work was informed by several previous experiences with curriculum development. In 2000 I developed structured teaching laboratories to support better acquisition of fundamental skills and iteratively refined both the content and the instructional design over the next five years, making pedagogical changes such as the use of pair programming and laboratory preparation sheets. These changes were evaluated through student surveys and formal experiments, leading to a co-authored course textbook, which was used from 2004 until 2014. In 2012 I introduced a graduate course in Computing Education - one of the few such courses in the world.

In 2014, I led a major revision of all three first-year courses to change the programming language and introduce alternate pathways into the major. I also addressed long-standing concerns about the level of feedback provided to students, by introducing an automated feedback system - 'CodeRunner' - to provide timely feedback for students. The evaluations of the changes to all three courses were positive, and CodeRunner has been adopted widely across the School as an automated assessment system.

Mentoring and Supporting Teaching

I have acted as a formal teaching mentor for four different academics in the SoCS, and informally for many others.

Andrew's way of teaching really inspired me at the time, and I have actually adopted a few of his techniques for my own teaching. I wish I had a lecturer like him when I was studying!

Ute Lörch, Lecturer, 2002

More than anyone else, Andrew provides teaching-related mentorship and guidance in our department. My office is next to Andrew's, and I regularly observe academics at all levels seeking Andrew's advice on a wide range of teaching and assessment matters.

Dr Paul Denny, 2017

After recognizing a lack of tutor training in the SoCS, in 2002 I developed a handbook for CS Tutors and a mentoring programme that consisted of monthly meetings to discuss pedagogies and pastoral care issues facing front-line tutoring staff. The program ran successfully from 2002–2007, after which the Faculty began formalizing tutor training. I continued to be involved in the delivery of the formal tutor training for CS tutors (2012–2015), and now provide mentoring on an informal/advisory basis.

He provides excellent mentoring to Limited-Term Tutor staff associated with his courses, and is, on his own initiative, extending his mentoring approach across all LTT staff employed within the department. This is typical of Andrew: solving a teaching problem within his own scope of operation and then looking to see how the best practice he has developed can be applied elsewhere. He is regularly consulted by colleagues who see him as a source of teaching inspiration.

Professor John Hosking, HoD, 2003

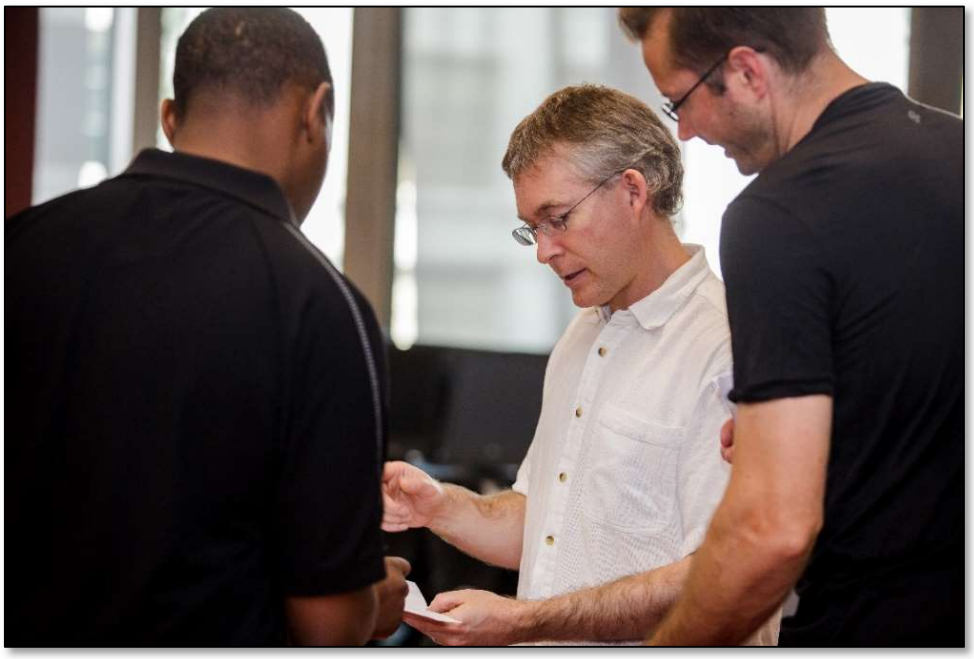
In 2016, I initiated a bi-weekly Teaching and Learning in Computer Science seminar series where we discuss teaching practice and curriculum issues. I coordinate the series and deliver several talks each semester. This initiative has raised the profile of teaching scholarship within the SoCS and developing a more engaged teaching community of practice. I have also established an expectation of teacher peer observation within the SoCS to better share and disseminate teaching practices between staff.

Andrew has also created a group that meets weekly for academics and graduate students to discuss teaching, learning and assessment-related matters ... with regular attendance by up to ten students and staff. [He also] organizes regular departmental teaching and learning seminars which have an even broader impact. Recently, this series has had expert guests from Chemistry discussing approaches for increasing student engagement in large classes, from Physics presenting their innovative studio-

based teaching and learning approach, from Education describing modern approaches to analysis of test questions, and kaiārahi from the Faculty of Science exploring ways in which we can embed social and environmental responsibilities across our major.

Dr Paul Denny, 2017

I have also been invited to deliver several seminars and workshops on teaching and learning activities to the wider University of Auckland community, including for the Department of Physics, Centre for Continuing Education (CCE), Centre for Learning and Research in Higher Education (CLeaR), and Library staff.



Evidence of Effectiveness

Community and Service Courses

From 1998 to 2004 I taught a component of a school holiday programme at UoA in which students aged 13–16 would spend a week learning to use a programming language to create 3D images or web pages. Feedback was informal, but extremely positive.

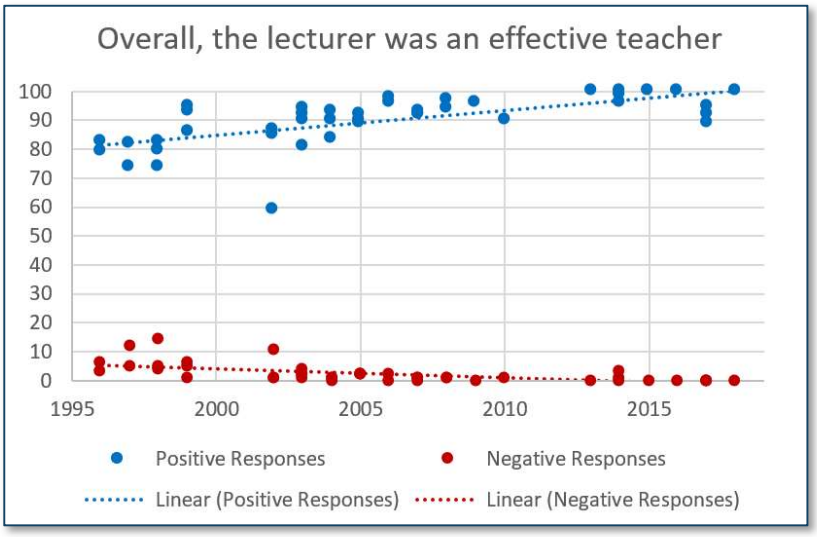
Between 2006 and 2010, I delivered several courses to adult students learning computing skills, administered through CCE. Students responded very positively to the courses, which were designed to provide practical skills to novices.

Table 1: Feedback for CCE Courses 'Overall the tutor was an effective teacher'

CCE Course	Year	Positive (%)	Negative (%)
Excel	2006	100	0
MS Access	2006	100	0
PowerPoint	2006	100	0
Excel	2009	100	0
PowerPoint	2009	100	0
Excel	2010	100	0
MS Access	2010	91	0
PowerPoint	2010	90	0

Formal UoA Courses

42 courses I taught were evaluated between 1996 and 2018, with responses from 4948 students.



- Since 1995: 4948 responses. 4368 positive (88.3%). 143 negative (2.9%).
- Since 2003: 2469 responses. 2300 positive (93.2%). 30 negative (1.2%).
- Since 2008: 730 responses. 695 positive (95.2%). 9 negative (1.2%).

I believe that the progressive improvement in my student evaluations is a result of my scholarly approach towards teaching, and my systematic reflective professional practice.

Table 2: Sample of results - 'Overall, the lecturer was an effective teacher', 1996-2018

Course	Year	Mean Likert (/5)	Positive (%)	Negative (%)
415.111.FC	1996	-	83	6
415.111.SC	1997	-	82	5
415.111.FC	1998	-	83	5
415.220.SC	1999	-	95	1
COMPSCI.101.S2	2002	4.2	87	1
COMPSCI.101.S2	2003	4.2	90	2
COMPSCI.101.S1	2004	4.5	93	1
COMPSCI.111.S2	2005	4.5	92	2
COMPSCI.111.S1	2006	4.8	98	0

Course	Year	Mean Likert (/5)	Positive (%)	Negative (%)
COMPSCI.111.S1	2007	4.5	93	1
COMPSCI.111.S1	2008	4.7	97	1
COMPSCI.111.S1	2009	4.7	96	0
COMPSCI.111.S2	2010	4.5	90	1
SOFTENG.250.S1	2013	4.7	100	0
COMPSCI.105.S2	2014	4.8	99	1
COMPSCI.107.S1	2015	4.7	100	0
COMPSCI.747.S2	2016	4.8	100	0
COMPSCI.107.S1	2017	4.5	95	0
COMPSCI.107.S1	2018	4.8	100	0

Academic Success of Graduate Students

Masters and PhD students are expected to produce work that is publishable in international venues, but I am especially pleased when students working on Honours projects or in taught courses can be nurtured to produce work of publishable quality.

Between 2013 and 2017, I mentored six of the students in the COMPSCI 747 Computing Education course in converting their coursework into research papers published in international conferences. Three Bachelor of Engineering (Honours) projects have been published, including one in a highly ranked international conference (ITiCSE).

Two of my students have been awarded a Best Student Paper award at an international conference, and I have supervised four BE(Hons) projects awarded industry prizes, all for projects that built educational tools for students (e.g. a game-based music learning app).

However, the most satisfying success story comes from a colleague who reports talking to a graduating student whom I taught many years earlier in a first-year course.

> *At the time I thought that he was a good lecturer, but I had not actually realised how well he had taught me to think, and that was invaluable, since it was really what helped me succeed in the end.*
>
> *Anonymous graduate student*

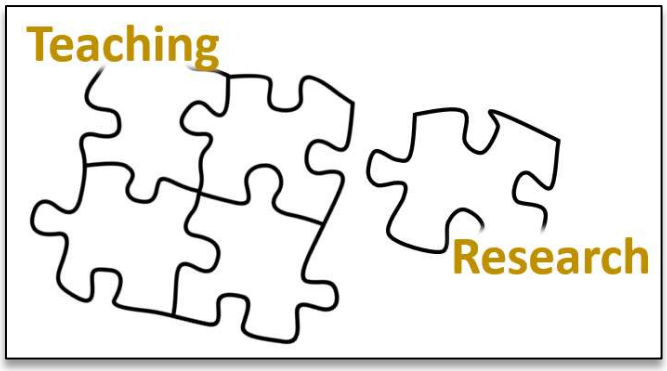
Teaching Scholarship and Professional Development

I consider my colleagues to be my most valuable resource in improving my own teaching practice, and credit them with the success I have had as a teacher. My daily discussions with colleagues provide invaluable insights. More formally, I have participated in more than 40 professional development workshops on teaching and learning; attended 20 International conferences on CS Education; completed the University's Postgraduate Certificate in Academic Practice; and was awarded a CLeaR Fellowship in 2014. Through these activities, and others, I am continuously refining the craft of teaching. In 2017 I received a Faculty of Science Teaching Excellence Award, followed by a University of Auckland Teaching Excellence Award in 2018.

Despite being an obviously gifted teacher, Andrew has continued to attend CPD courses on teaching methods.

Associate Professor Peter Gibbons, HoD 2002

Teaching and research are complementary and provide amazing synergy when they fit together and inform each other. Applying a scholarly approach to my teaching practice has resulted in more than 80 published academic papers about learning and teaching, including seven award-winning papers, and I lead a Computing Education and Learning Technology Group that meets weekly.



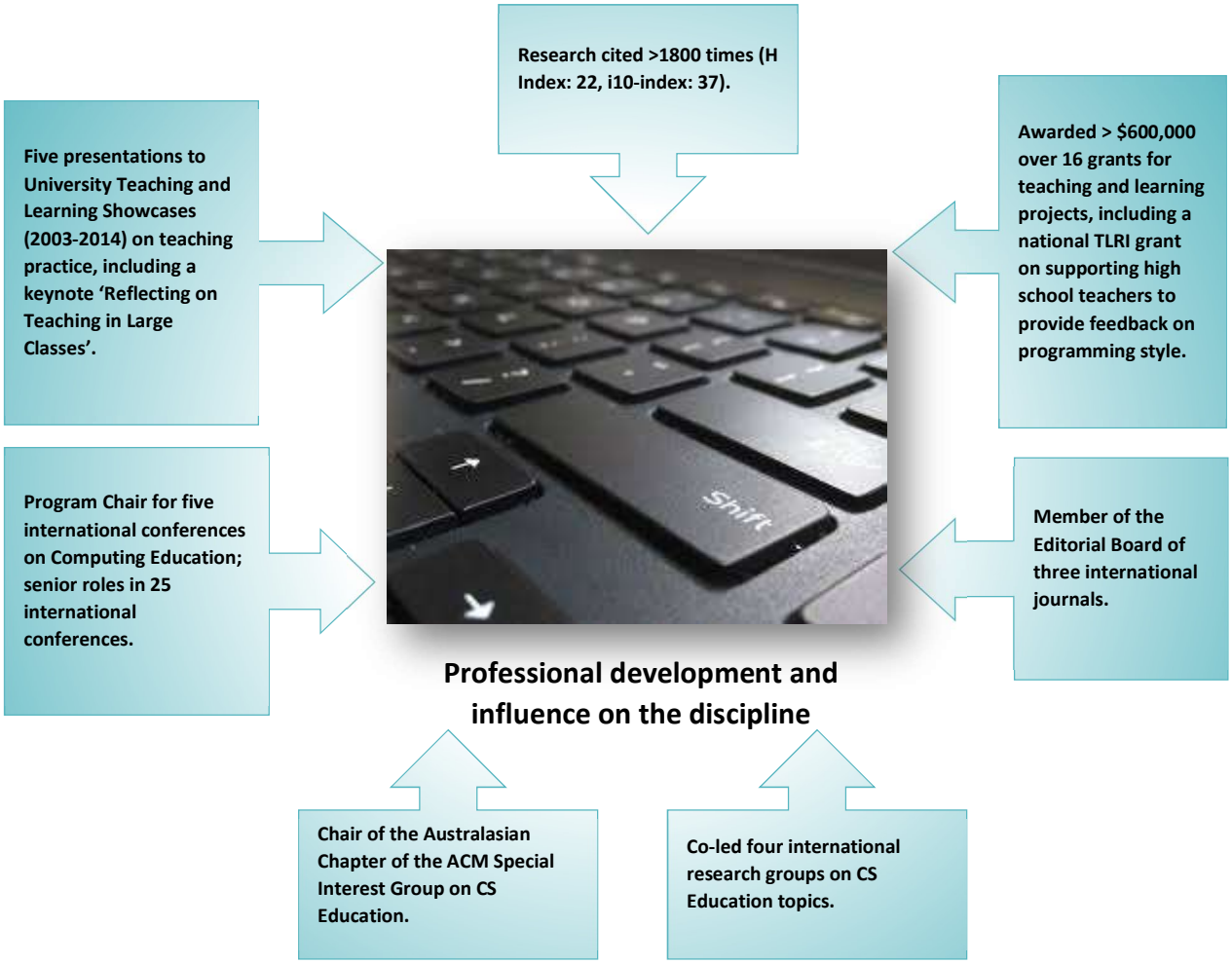
The three of us […] are focusing on the topic of in-flow peer review … We're writing because we would very much value your insights and wisdom on this topic.

Professor Kathi Fisler, Professor Shriram Krishnamurthi and Joe Politz, Brown University, 2014

In 2016 I received a Best Presentation award for the paper 'Learning to Program is Easy' at the ACM SIGCSE International Conference on Innovation and Technology in Computer Science Education for challenging the prevailing CS community view that it is difficult to learn programming.

I enjoyed reading the paper and recommend it - even though I disagree with his conclusions.

Professor Mark Guzdial, Georgia Tech, 2016



I enjoyed your workshop last week immensely. It was pitched perfectly for my learning needs, and your system of worksheets is exactly what I needed to nudge my teaching skills forward a bit, and my understanding of what good Python style looks like.

Digital Technology Teacher, 2018

Future Directions

Despite my extensive experience as an educator and education researcher, I feel acutely conscious of how little evidence we have for effective teaching practices in CS, and I intend to continue to evaluate and disseminate knowledge about teaching computing.

I am building a culture of reflective scholarly practice among the staff in the SoCS through seminars, peer observations, and collaborative scholarly publications that highlight the innovative practices of my colleagues, with the aim of normalizing such activities.

Further, I hope to embed ideas of ethical practices and social value across our tertiary computing curriculum and promote such values for students within the broader digital technologies curriculum.