

Part IV — Case Studies

Pawel Orzechowski

Edinburgh Medical School,
University of Edinburgh

Ozan Evkaya

School of Mathematics, The
University of Edinburgh

Learning is exciting. That's because every next thing we learn is new, yet it somehow connects with what we already knew in unexpected ways. The same premise makes teaching exciting – each course, each lesson, becomes better and gets rediscovered each time we deliver it. As teachers we come up with new examples, or tools, or metaphors, to entertain (and educate) our students.

Unfortunately, it takes a lot of work to create or change our learning materials. It may take months from first having an idea to validating if it works with real students. Many of us take years to hone a specific teaching method or course format, as you've read in other chapters. But think about this cheeky idea: it's great to learn from our mistakes, but it's even better to learn from someone else's mistakes! Similarly, while it's inspiring to create fantastic new teaching experiences, there's also a lot of joy in reading about other people's adventures. Especially if they tell us about their journey, not just the end product – we can vicariously relive their struggle and their joy. We learn from their journey.

Case studies you will find below tell stories of challenging what it means to teach programming. They question where, how and why students can engage with code. You will find a diversity of course formats, delivery methods, subjects, and teaching styles. You will read about the nitty-gritty of small experiments and the epic logistics of big initiatives. We are very excited to take you on this journey, because... teaching is exciting!

Theme — Building Learning Communities and Resources

A Gentle Introduction to Coding (Orzechowski 2026)

Here's an example of how the love of coding can spread to people who never tried it before. Authors describe an approach to learning programming skills by building on student's previously existing thinking patterns, and how it could be used to run a course. A group of 20 brave social scientists spent a week of their summer to pair-program together, exchange metaphors, and add coding to their toolkits.

This chapter is for you if you are curious about how lofty, idealistic ideals (of 'gentle' teaching) can be translated into course design and teaching methods. It also includes a lot of practical advice about feedback gathering, course design, and the practicalities of running a bootcamp.

The Wild(Life) Side of Pair Programming (Sepúlveda 2026)

A fascinating story about a group of students from Scotland's Rural College who pair-analysed environmental data in Excel. Most of students did not join the BSc Wildlife and Conservation Management to learn coding or data analysis. But they will need those skills in their daily work of maintaining natural environments. The author describes a journey of searching and testing various ways to excite the class, which then culminates in cross-pollinating of ideas a buzzing classroom.

This chapter is for you if you want to be inspired by how a simple change in a teaching method can transform the classroom - especially when the teacher emphatically customises it for the specific audience.

Interdisciplinary by Design (Michielin and Otty 2026)

To create a sustainable training ecosystem it takes planning, skills and determination, as witnessed by the story of The Centre for Data, Culture, & Society (CDCS). This initiative was created within the University of Edinburgh to address the skills gap amongst non-technical researchers from diverse academic backgrounds. Over the years, a generation of scholars at different career stages used CDCS's training programs to enhance their digital literacy. The chapter showcases specific training materials and activities, from silent-disco to focused masterclasses.

This chapter is for you if you are looking for different training approaches that target scholars from non-technical backgrounds. You might get inspired by specific teaching materials, teaching methods or the very way in which CDCS serves its purpose. You'll also find useful insights if you're planning to start or rejuvenate a training programme in your own organisation.

Changing Civil Engineering Students' Mindset Toward Programming (Chien 2026)

Ponder this question: does being an expert in one area (e.g., civil engineering) change the way you program? The author describes what patterns they noticed among civil engineering students in terms of problem solving and code writing. You'll see how skills of connecting physical structures, logic and math get translated into code. Is their impact visible even in high level activities like code reuse, generalisation, and problem decomposition? It is not just about the differences in applications, but also about finding a golden middle ground between high-level and detailed, between applied and abstract.

This chapter is for you not just if you work with civil engineers, but also if you are an instructor in a field with its own specific ways of problem-solving. You might also enjoy pedagogical parallels woven from jumping between niche examples for engineering students, and high-level course design.

Lost in Translation (Konovalov, Michielin, and Yanina Bellini Saibene 2026)

Three educators combine their experiences of translating (and coordinating teams who translated) teaching materials from English into Ukrainian, Italian, Spanish and other human languages. They provide an actionable blueprint of how to start, run and maintain big translation projects, especially in the context of teaching coding. How do you make sure all translators are on the same page? What concepts do you translate and which do you leave in English? How do you ensure good quality of translations and motivate teams of translators (who are frequently volunteers)? This chapter has all the answers. Spoiler alert: it is all about process, source control, and synchronising expectations within the team.

This chapter is for you if you take part in (or manage) any translation projects, even if only tangentially connected to coding. Indeed, you will learn a lot about: coordinating teams; shaping quality-assurance processes; and future-proofing content. This chapter shows that anything worth doing, deep inside, is about project management and empathy.

References

Chien, Nguyen. 2026. "Changing Civil Engineering Students' Mindset Toward Programming." In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Konovalov, Olexandr, Lucia Michielin, and Yanina Bellini Saibene. 2026. "Lost in Translation: Complexities and Good Practices of Translating Coding-Based Teaching Materials Across Multiple Human Languages." In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Michielin, Lucia, and Lisa Otty. 2026. "Interdisciplinary by Design: The Centre for Data, Culture, & Society Training Programme." In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Orzechowski, Pawel. 2026. "A Gentle Introduction to Coding: Narratives from a Programming Summer School for Social Sciences and Humanities." In *Teaching Programming Across Disciplines*. Edinburgh Diamond.

Sepúlveda, Ignacio. 2026. "Case Study from Scotland's Rural College: The Wild(life) Side of Pair Programming." In *Teaching Programming Across Disciplines*. Edinburgh Diamond.