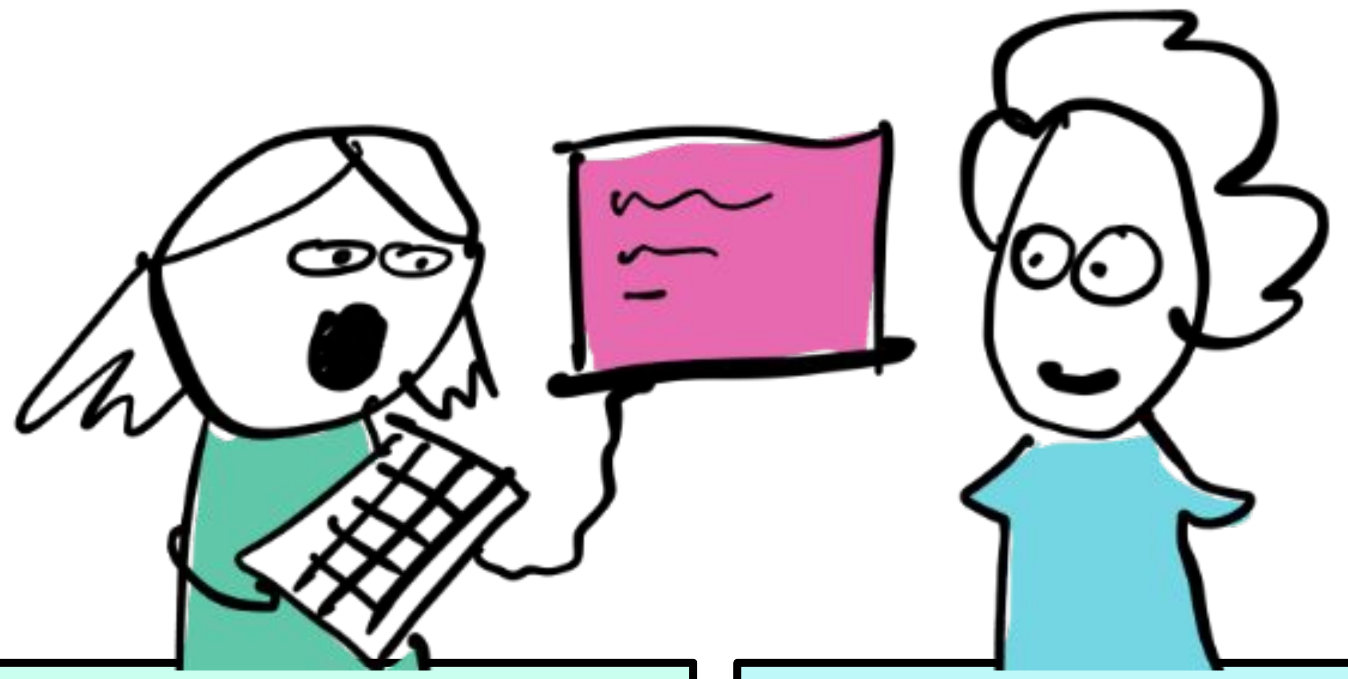
Fiona McNeill, Reader in Computer Science Education, School of Informatics
Pawel Orzechowski, Senior Teaching Fellow in Programming for Business, Business School

THE UNIVERSITY *of* EDINBURGH
informatics
UNIVERSITY OF EDINBURGH
Business School

Watch Our Video:
https://media.ed.ac.uk/media/t/1_c5gpv4e5

Pair Programming - two students work on one problem, taking turns to type, explain and ask questions

Two Roles in Pair Programming and How It Works



the Driver = active person, problem-solving, thinking out loud

the Navigator = reflective person, looks out for bugs, is a Rubber-Duck*

They switch roles every 10 minutes

Students learn from each other not just the course material, but also thinking patterns and problem-solving techniques.

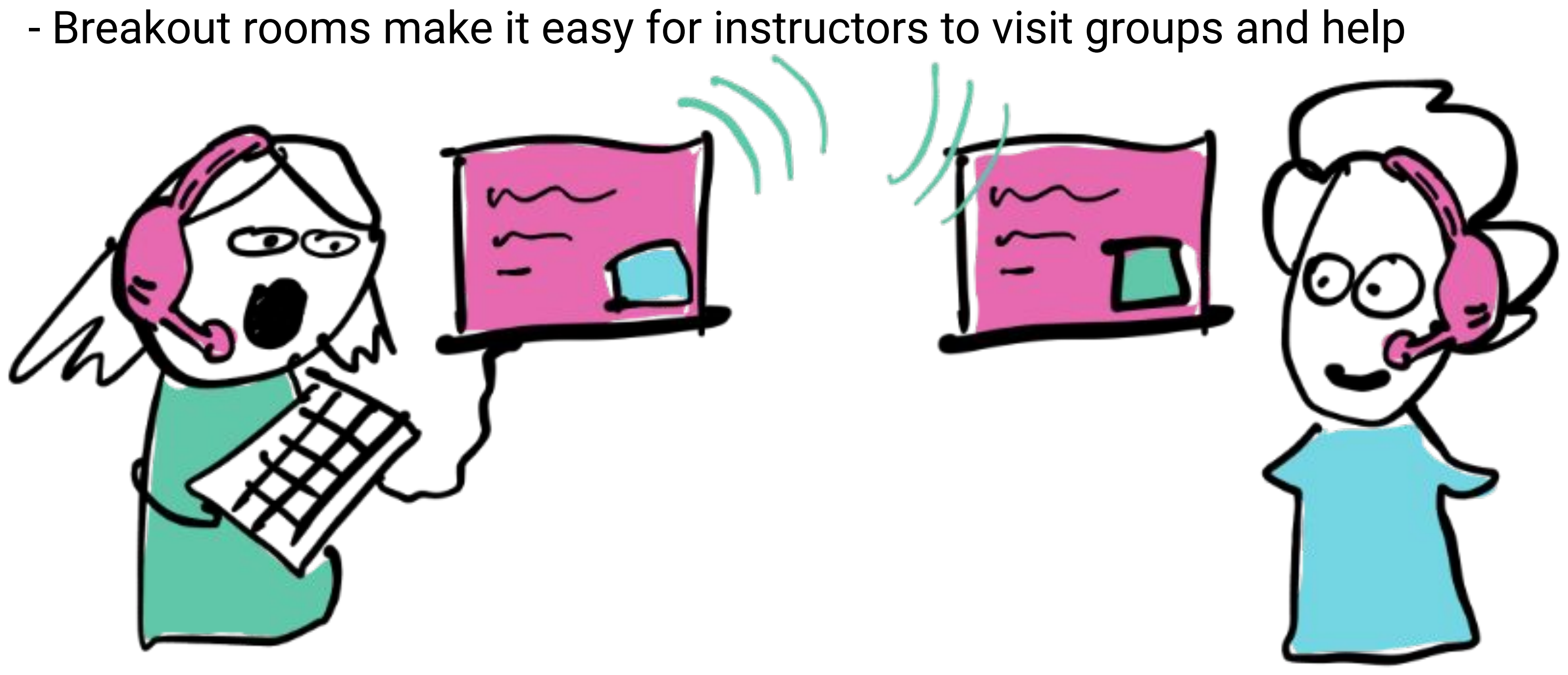
For each pair the goal is to not just arrive at the solution together but to also achieve the learning goals together. They learn to justify and explain their shared code, and are often challenged by the questions and answers of their partner.

Pairs are randomised between tutorials, so that students can meet and work closely with a number of their colleagues.

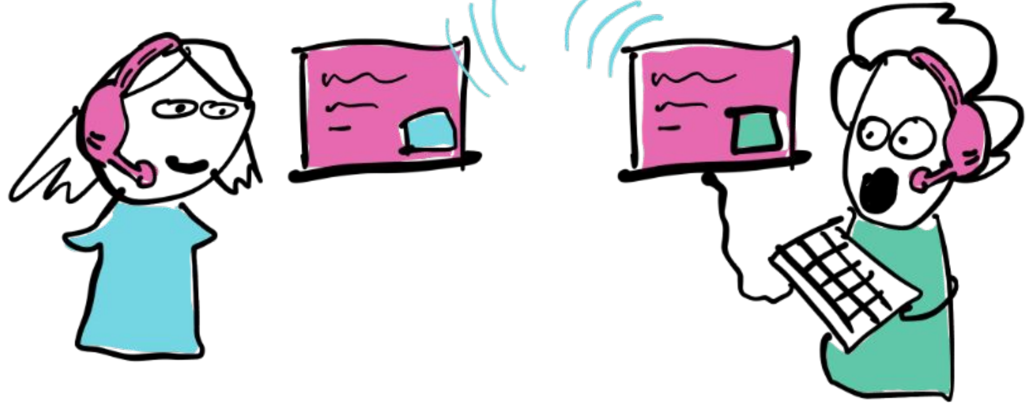
This creates a **feeling of community** (valuable especially during online delivery) and enables students how to work with people of different personalities, backgrounds and skill levels.

An instructor visit each group a few times during the tutorial or whenever someone raises a hand. Their role is to ensure that students stick to their roles and remember to rotate who's driving. Instructor also helps with technical queries and good practices.

In an online scenario:
- Screen sharing replaces sharing one computer and keyboard
- Code lives on the cloud, so both students work on the same solution
- Video call enables constant conversation and discussion
- Breakout rooms make it easy for instructors to visit groups and help



They switch roles every 10 minutes



Courses We Used Pair Programming on

Business School, UG & PG, Intro to Python Programming
- 2 years (1 year in person, 1 year online), 4 x 70 students

Informatics, UG, Object-Oriented Programming
- 1 Year (online), 400 students. (See student quotes below)

We've seen positive impact on community and learning, but also witnessed the importance of teaching support and of balancing difficulty levels. Courses had learners at **levels from total novices to very experienced**. Divisions can grow in such a course, with those with little experience at risk of believing they are no good at programming, when they simply lack practice.

Pair Programming in Education

"...the benefits of pair programming include increased success rates in introductory courses, increased **retention** in the major, higher quality software, higher student **confidence** in solutions, and improvement in **learning outcomes.** Moreover, there is some evidence that women, in particular, benefit from pair programming."[1]

[1] B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy and C. Zander, "Pair programming in education: a literature review," in Computer Science Education, vol. 21, no. 2, pp. 135-173, 2011, https://doi.org/10.1080/08993408.2011.579808.

***Rubber-Ducking** = Describing your problem to someone who's listening actively, asking clarifying questions.

The legend says that simply explaining your problems to a rubber duck will enable you to solve them by yourself. Because you only *really* understand something if you can explain it to someone else.

Student Feedback and Space for Improvement

Positive student feedback highlights:

- learning from each other's coding style and problem solving
- higher engagement and enjoyment
- benefits of social interaction.

"The lab session pair programming gave us some practise so we could improve our skills, and allowed for some socialising with a few peers. On top of that, it introduced me to a surprisingly good style of programming I never would've come across before the course."

"I enjoyed the pair programming sessions a lot. It was personally much more engaging and helpful"

"I liked pair programming because of the chance to interact with other people."

Negative student feedback highlights the importance of:

- clarity of the pair roles, switching and working together
- training the demonstrators to facilitate the exercise well
- resistance and worry of students towards working with people at a different level from them (both higher and lower)