Using a Virtual Computing Lab to Teach Programming at a Distance

Phil Hackett
School of Computing & Communications, The Open University, UK
phil.hackett@open.ac.uk

Michel Wermelinger
School of Computing & Communications, The Open University, UK
michel.wermelinger@open.ac.uk

Karen Kear
School of Computing & Communications, The Open University, UK
karen.kear@open.ac.uk

Chris Douce
School of Computing & Communications, The Open University, UK
chris.douce@open.ac.uk

ABSTRACT

This paper discusses a pilot research project, which investigated the use of online collaborative IDEs (Integrated development environments) during a first-year computing degree course. The IDEs used can be described as virtual computing labs because they replicate some of the actions possible in physical computing labs. Students were supported by a tutor with real-time help and feedback provided, whilst they were programming, without being collocated. The use of two different platforms is considered with the benefits and drawbacks discussed. Students and tutors indicated that they would like to use a virtual computing lab approach in the future.

CCS CONCEPTS

• **Collaborative and social computing systems and tools**; • **Distance Learning**; • **Applied computing**; • **Education**;

KEYWORDS

novice programmers, collaborative programming, virtual computing laboratory

ACM Reference Format:
Phil Hackett, Michel Wermelinger, Karen Kear, and Chris Douce. 2023. Using a Virtual Computing Lab to Teach Programming at a Distance. In *Computing Education Practice (CEP '23), January 06, 2023, Durham, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3573260.3573262

1 INTRODUCTION

Supporting students to develop their programming skills at a distance is a challenge faced by many educational establishments both nationally and internationally. Students face significant barriers when they cannot learn with a face-to-face teacher or peer support in a physical computer laboratory.

CEP '23, January 06, 2023, Durham, United Kingdom

ACM ISBN 978-1-4503-9821-3/23/01…$15.00
https://doi.org/10.1145/3573260.3573262

The approach used in this paper could be useful in a distance learning context and several other scenarios relevant to computing education practitioners.

One example is remote CPD for teachers who are upskilling so that they can teach programming in UK schools. It is not always possible, or necessary, for an expert teacher of programming, to deliver CPD face-to-face. Using a virtual computing lab, would facilitate better supported remote CPD for teachers who are looking to upskill in programming.

Another example is in providing remote access to expert teachers of programming, to students that may not have access to specialist teachers, such as students who are in an alternative provision environment like a pupil referral unit or hospital.

1.1 Previous Research

It has been recognised that collaborative learning platforms [1] could be an effective aid for novice programmers. Carter et. al. [2] suggest that social factors heavily contribute to learners' eventual success or failure, with Xinogalos at. al. [3] finding that collaborative problem solving helps students to be more confident. Maguire et. al. [4] investigated a range of factors with students starting introductory programming classes including sociodemographic factors, engagement, social media usage and many others – they found that confidence in programming is an independent predictor of exam performance when learning to program.

In research carried out by Sentence & Waite [5] with school aged learners, use of a 'shared example program' is seen as key when a teacher is talking about programming. They cite numerous sources, including [6–8], to describe the positive impact that talking, and peer instruction has on learning outcomes.

1.2 Context

The context of this research is with distance learning students studying TM112 'Introduction to computing and information technology 2'. This is a first year, 30 credit, undergraduate computing module, which ran between April and September 2022. It is the students' first computing module with a text-based programming language, and follows on from TM111, which introduces programming using a visual language.

The programming aspect of the module makes up one third of the material alongside other computing topics such as binary data, computer architecture and ethics. Assessment is via two assignments

and an end of module assessment, which includes a Python-based project.

We have carried out this research to investigate recent developments in innovative technologies that facilitate student-tutor and student-student collaboration whilst programming. A key part of the research is to identify the feasibility of using these technologies during distance learning tutorials, as well as gathering student and tutor perception of this approach.

2 PILOT RESEARCH PROJECT

The pilot research project focused on the synchronous online programming tutorials which students attend with a tutor. Three different methods of delivering online programming tutorials are described below.

One approach, currently used, involves the students using an 'integrated development environment' (IDE) on their own computer and involves no collaboration and limited tutor support. Any attempts at programming by students are not visible to the tutor unless a student shares their screen to receive feedback.

An alternative approach involves the use of a shared collaborative programming environment which allows tutors to interact with students' code as the student is coding in an online IDE (https://replit.com). This approach requires use of additional video conferencing software, such as Adobe Connect, to enable audio and video for discussion.

Another approach involves the use of a single platform (https://codingrooms.com), which includes integrated video conferencing. This software enables an overview of multiple students programming at the same time, with tutors able to interact with and support individual students or groups in real-time.

With REPLIT or Coding Rooms, each student uses their own online IDE which the tutor can see in real-time – feedback and discussion could happen immediately. Students are also able to collaborate in these coding environments, outside of tutorials if they wish. Both REPLIT and Coding Rooms can be considered virtual computing laboratories.

2.1 Research procedures

We used a mixed methods approach to survey students and ran a focus group with tutors; the first author also observed the REPLIT and Coding Rooms tutorials, noting the strategies used during the virtual computing lab sessions, specifically:

- Social interactions
- Technology utilisation
- Collaboration

2.1.1 Pre-survey. At the start of the April 2022 presentation of TM112, we invited students to participate in this pilot research project. Students were provided with an opportunity to attend additional programming tutorials, using either REPLIT or Coding Rooms. To participate, students needed to complete a questionnaire which captured prior programming experience and measured their confidence in programming, prior to any intervention. This was done using an adapted version of the 'computing attitudes' survey, which is a validated instrument [9]. Students also had an option to choose whether they would like to attend virtual computing lab sessions using either REPLIT or Coding Rooms.

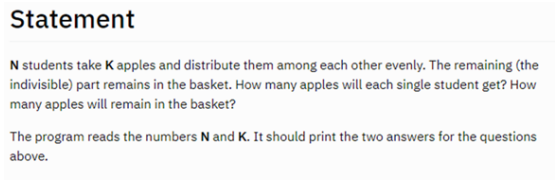

Figure 1: Programming problem statement

Fifty-two students completed this first survey, with 15 requesting an invitation to attend REPLIT sessions and 32 opting for Coding Rooms. This enabled three groups to be created with REPLIT students, Coding Rooms students and a control group.

2.1.2 Virtual computing lab sessions. Dates for the virtual computing lab programming sessions were arranged with tutors, and these were advertised to students, via email, throughout the duration of the TM112 module. This was different to the normal method of accessing tutorials, which is via a booking system on the module website. This may have contributed to the low turnout to some of the sessions because students were only aware of the sessions via email reminders - they did not have access to a list of prebooked events, which would happen with normal tutorials. The largest number of students at a session was 6, with most sessions having 2 or 3 students.

During the sessions, it was observed that tutors were able to help students to overcome common issues which prevented progress, such as naming variables incorrectly or missing a matching parenthesis within their code.

Tutors were able to identify misconceptions quickly and support students to overcome issues. An example of this is when one student was attempting to solve the following problem statement:

The tutor could see that the student first tried to use a ***for*** loop then a ***while*** loop – the tutor was able to intervene to remind the student to consider the inputs and outputs needed. The student was able solve the problem with modulo division, as expected, once they had thought about the problem and followed the problem-solving process taught on TM112.

Without intervention from the tutor, the student may not have been able to complete this task and may have been left feeling frustrated and lacking in confidence.

2.1.3 Using a virtual computing lab. Students who used REPLIT were required to join a video conferencing session and then click on a link to the online IDE provided. Within the IDE, students were presented with tasks, such as that in Figure 1. Tutors were able to observe and interact with any students' IDE. Only one IDE could be viewed at any one time using REPLIT. Communication with the students was via microphone through the separate video conferencing software used (Adobe Connect).

Students who used Coding Rooms could see similar activities to the REPLIT students, but the video conferencing facility was built in and considered to be of better quality by the tutors. Another advantage to tutors was that the IDE of all students could be seen at the same time using a variable sized grid view of 4 or 6 IDEs.
One tutor, who used both platforms, considered REPLIT to be more intuitive ('I just had everything I felt I expected'), but the most

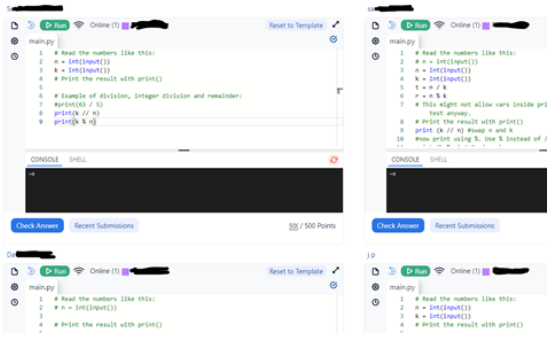

Figure 2: Coding Rooms - grid view of students

useful feature was in Coding Rooms, which provided a grid view (Figure 2).

Both platforms allow the use of automated testing of tasks using unit or input/output matching, although limited use of this was made during this pilot research project.

2.1.4 Post-survey. At the end of the module, we sent students the computing attitudes survey again and also asked them to provide comments on their perception of using virtual computing lab environments in their tutorials.

Student perceptions of the use of virtual computing lab environments is discussed in Section 3. Feedback from tutors was also gathered in a small focus group with their comments discussed in Section 4.

3 RESULTS

The student response rate was disappointingly low. Of the 16 students who answered both surveys, 4 had attended REPLIT and 2 had attended Coding Rooms sessions (students 4 & 5 in Table 1). The other 10 students attended no programming tutorials at all.

The computing attitudes survey that was used, included questions relating to personal interest, programming, and fixed mindset. In analysing the results, we focused purely on the questions relating to programming and compared pre- and post-survey results.

The change in confidence column indicates the difference between the score from the first survey and the score in the second survey.

The average change in confidence was slightly greater in the students attending virtual computing lab sessions (1.25) than in students who did not attend the sessions (0.8).

An interesting observation from these results is that students who already had a higher qualification had the greatest decrease in their programming confidence, as shown in Table 1 and Table 2.

Although the students who attended virtual computing lab sessions showed a greater increase in average confidence, considering the small number of responses, there is not enough data to draw firm conclusions.

In answering questions relating to the use of a virtual computing lab environment in the future, students 1-6 indicated that they would like to use this approach in future modules that involve programming. These students also stated that they would like to collaborate with other students whilst learning to program, using either REPLIT or Coding Rooms.

Students 1-6 were also asked if they thought their understanding and enjoyment of programming had improved, as a result of attending the sessions. All students, apart from Student 6, answered 'yes' to both questions.

An interesting observation is that student 6 indicated that they would like to use a virtual computing lab in the future, even though they had a negative experience and did not enjoy the session they attended.

4 DISCUSSION

The use of a virtual computing lab platform to teach programming has not been attempted previously with TM112 students. Tutors have taught Python programming during day-school sessions, which were face-to-face (pre-pandemic), but most programming sessions have been delivered online, without significant student interaction or collaboration. This new approach to teaching programming, required tutors to take a different approach to their normal tutorials.

The tutors involved in this pilot research project reflected on their previous approach, which gave limited opportunity to see what students were doing and to catch misconceptions as they happened. Whilst this is possible in a face-to-face scenario, it would not usually be possible whilst teaching programming at a distance. The positives identified by tutors included the ability to 'see over the shoulder' of students. Watching students develop an answer to a programming problem and observe how they responded to syntax, run-time or logical errors gave insight into student understanding that was previously not possible.

Another positive was the fact that both platforms allowed students to work at their own pace, rather than at the pace of a tutor, which would be the case in a usual tutorial. One tutor commented that, by using a virtual computing lab, they had realised some

Table 1: Change in confidence – attended sessions

Student	Prior Education	Sessions Attended	Change in Confidence
Student 1	A-Levels or equivalent	3+	4
Student 2	GCSE or equivalent	3+	2
Student 3	A-Levels or equivalent	2	1
Student 4	Higher national certificate	1	0
Student 5	GCSE or equivalent	1	0
Student 6	First degree of UK institution	1	-2

Table 2: Change in confidence - no sessions attended

Student	Prior Education	Change in Confidence
Student 7	A-Levels or equivalent	3
Student 8	A-Levels or equivalent	3
Student 9	GCSE or equivalent	3
Student 10	GCSE or equivalent	2
Student 11	A-Levels or equivalent	1
Student 12	A-Levels or equivalent	1
Student 13	GCSE or equivalent	1
Student 14	GCSE or equivalent	-1
Student 15	First degree of UK institution	-2
Student 16	UK Doctorate degree	-3

students were less confident and able than anticipated – they recognised that their previous tutorial approach involved concepts that were too difficult and a pace that was too quick for some students.

Tutors and students identified some potential issues with the approach used. The main issue was the lack of 'private' audio – when using the microphone to provide support, all students could hear the tutor, which might have been distracting and embarrassing. Although during this pilot the student numbers were small (2 – 6 students per session), tutors indicated issues with scaling to bigger groups. These issues could possibly be mitigated by using breakout rooms and the tutor moving between rooms to provide support.
The overwhelming response from tutors though, is that the use of a virtual computing lab to support students, while they are programming, is a good thing and worth exploring further.

4.1 Who else has done this?

Whilst there has been previous research relating to the use of this type of technology with pair-programming [10] and the use of feedback in a face-to-face scenario [11], there is limited research on tutor-student feedback and support, whilst teaching programming to novice programmers at a distance.

4.2 Next steps

Further research into the use of virtual computing lab environments will be carried out on future distance learning modules. To capture more representative data, students participating will be asked for feedback immediately after attending sessions, instead of at the end of the module.

There will be a greater emphasis on facilitating student-student collaboration as well as tutor-student support. Developing an approach that works with larger groups of students will also form part of further research.
Further exploration of the built-in testing tools, which provide the use of the automated testing and feedback, alongside real-time feedback, as described by [11] is also a consideration in further research.

5 CONCLUSIONS

Using virtual computing lab technologies enables distance learning tutors to provide real-time feedback and support to students in a way that was not previously possible. The two tools investigated, REPLIT and Coding Rooms, appear to be suitable for providing feedback and support to students learning to program at a distance.

Students and tutors responded positively with most participating students stating that the sessions improved their understanding and enjoyment of programming.

The computing attitudes survey results suggest that students who participated in the sessions may have had a greater change in average confidence levels than students who did not attend the virtual computing lab sessions.

REFERENCES

[1] Omer, U., Farooq, M. S. and Abid, A. (2021) 'Introductory programming course: review and future implications', PeerJ. Computer science, 7, p. e647. doi: 10.7717/peerj-cs.647.

[2] Carter, A.S., Hundhausen, C.D. and Adesope, O., (2017). Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. ACM Transactions on Computing Education (TOCE), 17(3), pp.1-20. https://dl.acm.org/doi/abs/10.1145/3120259

[3] Xinogalos, S. Satratzemi M, Chatzigeorgiou A, Tsompanoudi D. (2019) 'Factors Affecting Students' Performance in Distributed Pair Programming', Journal of educational computing research, 57(2), pp. 513–544. doi: 10.1177/0735633117749432.

[4] Maguire, P., Maguire, R., & Kelly, R. (2017) Using automatic machine assessment to teach computer programming, CS Ed., 27:3-4, pp. 197-214, https://doi.org/10.1080/08993408.2018.1435113

[5] Sentence, S. & Waite, J. (2021). 'Teachers' Perspectives on Talk in the Programming Classroom: Language as a Mediator.' Proc. of 17th ACM Conf. on Int. Computing Education Research (ICER 2021). ACM pp. 266–280. DOI: https://doi.org/10.1145/3446871.3469751

[6] Israel, M., Wherfel, Q., Shehab, S., Melvin, O. and Lash, T. (2017). Describing Elementary Students' Interactions in K-5 Puzzle-based Computer Science Environments using the Collaborative Computing Observation Instrument (C-COI). In Proc. 2017 ACM Conf. on Int. Computing Education Research (ICER '17). ACM, 110–117. https://doi.org/10.1145/3105726.3106167 00014.

[7] Porter, L. Bailey Lee, C. Simon, B. and Zingaro, D. (2011). Peer instruction: do students really learn from peer discussion in computing? In Proc. of 7th int. workshop on Computing education research. ACM, 45–52. http://dl.acm.org/citation.cfm?id=2016923

[8] Zingaro, D. (2014). Peer Instruction Contributes to Self-efficacy in CS1. In Proc. of 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14). ACM, 373–378. https://doi.org/10.1145/2538862.2538878

[9] Dorn, B. & Elliott Tew, A. (2015) Empirical validation and application of the computing attitudes survey, Computer Science Education, 25:1, 1-36, DOI: 10.1080/08993408.2015.1014142

[10] Adeliyi, A. Wermelinger, M. Kear, K. and Rosewell, J. (2021) Investigating Remote Pair Programming In Part-Time Distance Education. In Proc. of 2021 Conference on United Kingdom & Ireland Computing Education Research (UKICER '21). ACM, Article 6, 1–7. https://doi.org/10.1145/3481282.3481290

[11] Grawemeyer, B. Halloran, J. England, M. and Croft, D. (2022). Feedback and Engagement on an Introductory Programming Module. In Computing Education Practice 2022 (CEP 2022). ACM, 17–20. https://doi.org/10.1145/3498343.3498348