

# ILECTURES - EXPLORING THE INTERACTIVE LECTURE USING WEB-BASED BROADCASTING TECHNOLOGY

*John Busch, Philip Hanna & Neil Anderson  
Queen's University Belfast*

## Abstract

The continued use of traditional lecturing across Higher Education as the main teaching and learning approach in many disciplines must be challenged. An increasing number of studies suggest that this approach, compared to more active learning methods, is the least effective. In counterargument, the use of traditional lectures are often justified as necessary given a large student population. By analysing the implementation of a web based broadcasting approach which replaced the traditional lecture within a programming-based module, and thereby removed the student population rationale, it was hoped that the student learning experience would become more active and ultimately enhance learning on the module. The implemented model replaces the traditional approach of students attending an on-campus lecture theatre with a web-based live broadcast approach that focuses on students being active learners rather than passive recipients. Students 'attend' by viewing a live broadcast of the lecturer, presented as a talking head, and the lecturer's desktop, via a web browser. Video and audio communication is primarily from tutor to students, with text-based comments used to provide communication from students to tutor. This approach promotes active learning by allowing student to perform activities on their own computer rather than the passive viewing and listening common encountered in large lecture classes. By analysing this approach over two years (n = 234 students) results indicate that 89.6% of students rated the approach as offering a highly positive learning experience. Comparing student performance across three academic years also indicates a positive change. A small data analytic analysis was conducted into student participation levels and suggests that the student cohort's willingness to engage with the broadcast lectures material is high.

## Introduction & Background

Learning programming for software development is not perceived to be easy (Milne & Rowe, 2002). In many Universities the teaching of programming utilizes the traditional lecture as a means of introducing and exploring concepts. However, an increasing number of academics argue that the traditional lecture, or 'Sage on the Stage', with large numbers of students needs to be rethought. It is clear that the traditional lecture has certain advantages: time, immediate communication between and tutor and students (Reece & Walker, 2003; Biggs & Tang, 2007; Curzon, 2003). However research suggests that this traditional format of presenting information is not best suited to students studying programming (Robins, Rountree & Rountree, 2003). Exley and Dennick (2004) with as Biggs and Tang (2007) reported that lectures offer one-way communication, with students mostly being passive receivers of information, resulting in levels of student concentration dropping after 15-20 minutes. In turn leading to an ineffective learning experience for the students.

Fry, Ketteridge, and Marshall (2000) found that active students within the lecture environment are reported to have better knowledge capacity, alongside demonstrating a higher rate of transfer of knowledge and more effective problem solving ability than their passive counterparts. Recent studies (Anderson et al, 2007; Freeman et al, 2014; Georjoui & Sharma, 2015; McConnell, 1996) have found that *Active Learning* embedded into traditional lectures can improve the student's learning experience, with some studies (Sjostedt, 2015) reporting significantly improved student performance. However, mapping active learning paradigms onto traditional lectures with large students is difficult. Teaching programming may be even more difficult. Many active learning approaches discussed in academia focuses on short collaborative tasks that may be suitable to developing the introductory skills to programming. Developing small group and problem-based tasks to manage large numbers in a lecture

hall is one such approach (Wolfe et al, 2015). The growing body of research highlights that active students within lectures promote a richer learning experience that can lead to better performance on final grades.

The *Workshop* approach, mooted by Reece and Walker (2003), is defined as a learning session that gives students the chance to develop practical skills in a simulated situation by linking the theory with the skill. They argue that the approach has the ability to reinforce learning in a realistic and meaningful way. Additionally, Brown and Atkins (1988) suggest that practical skills are taught more effectively in a hands-on practical lab rather than in a lecture. In other words, programming can only be mastered by practicing it (Vihavainen, 2011). Gill and Greenhow (2007) also show this type of approach can improve student confidence levels in a subject often found to be difficult. While workshops can give students plenty of feedback on their 'doing' via tutors or their peers (Race, 2015).

The practice of *Lecture Casting*, defined by Zhu and Bergom (2010) as the recording of a lecture theatre using specific software and making the recording available electronically to download from the web. Recording lectures gives students the ability to replay or revisit the material (Boyle, 1997; Bonk & Zhang, 2008; Evans, 2008;). Giving the student the ability to revisit the complex topics again via a lecture recording can help better learn complex material that is unlikely to be fully assimilated when first encountered. Further research conducted by Fernandez, Simo, and Sallan (2009) found that students are able to access material after they were introduced to it via a live lecture. Being able to revisit the material gives the student the opportunity to check whether or not they have understood the material.

Viewing the lecturer within the recording can be an important mechanism to enhance the learning (Bonk & Zhang, 2008; Dey, Burn, & Gerdes, 2009). Students typically prefer to see the lecturer and her/his activity (using the blackboard, performing an experiment, etc.). Pale et (2007) states that students should see the lecturer's mannerisms because body language sometimes expresses more than words themselves. The live lecture recordings provide the opportunity for dynamic variations in vocal tone and pronunciation. These visual and audio combined cues can provide insight and emphasis on material that is more important. Fernandez, Simo and Sallan (2009) state that students indicated that there was a decreased information overload and an increase in information quality when lecture casting was implemented. Exley and Dennick (2004) purpose that live lectures a good mechanism to communicate enthusiasm for a subject area, which helps to promote a good learning experience for students.

Lecture casts research also comments that they are used as a revision aid for students to learn and understand complex processes (Jaki, 2009; Evans, 2008). As an aid for note taking (Zhu & Bergom, 2010), and to support different types of learners (Fernandez, Simo & Sallan 2009; Pale et, 2007). Additionally, tutors that are recording can use this technology for self-improvement through self-reflection of the recorded material (Hurst & Waizenegger, 2006; Odhabi & Nicks-McCaleb, 2009).

The above research advocates that to improve the teaching and learning of programming, traditional lectures need to support 'learning by doing' with the benefit of recording a visual copy. It must also keep hold of their particularly learning advantages. This study intends to explore a web-based broadcasting system that will introduce workshop interactivity into the traditional lecture whilst retaining the learning advantages.

## Research Objectives

This research study primarily examines and analyses the inclusion of a web-based broadcasting approach, entitled iLecture, that replaces a traditional lecturing slot within a full-year undergraduate programme of study. The system allows lecturers to broadcast a live traditional lecture, with students viewing the broadcast using standard web-based browser technologies. The analysis seeks to establish if the inclusion of the iLecture along with traditional lecturing techniques and an active learning

approach can enhance learner experience and performance. The study will make use of learner data analytics to enrich the qualitative results reported by the student survey, resulting in recommendations of best practices for live lecture broadcasting.

## iLecture System

The 'iLecture' is a browser-based broadcasting system that was surveyed across the 2013-14 and 2014-15 academic terms. It lets students view a live lecture cast in a portion of their screen whilst allowing them to follow along with the demonstration of code using their own IDE. This allows the students to be active participants in real-time programming and code development during the lecture. In Figure 1 the tutor is demonstrating and explaining code (on the right), and on the left the student can duplicate or follow along with the code in their own IDE. Two live video sources are broadcasted, one is the tutor's screen and the other is a medium close up. Audio from the tutor is also transmitted. A web-based text messaging service, used by students and tutor for transmission of queries, explanations, issues etc., is shown in the bottom right of Figure 1.

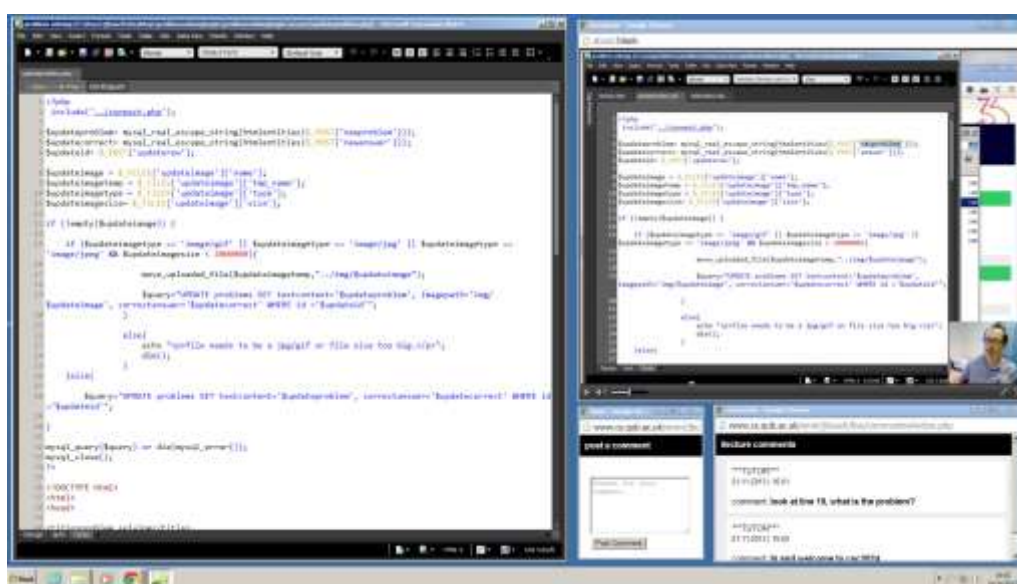


Figure 1. Typical student set-up for live broadcasting UI

The system encourages the student to be active rather than passive, allowing the student to practice coding with the tutor. The lag of the live broadcast was kept as low as possible, around 5 - 10 seconds was typical. As such, the transmission of text messaging between tutor and students was considered synchronous. In 2013-14, a third party cloud-based solution, Twitch.tv, was used to handle the streaming of the live lecture broadcast. The following year we implemented our own in-house streaming web server, based on Nginx HTTP server. This change did not alter the functionality of the system design but helped improve on the broadcasting and streaming performance. Additionally, there was also a slight superficial change to the layout of the web-based UI in 2014-15.

Each iLecture was timetabled for two hours per week during the teaching semester. Not all weeks included an iLecture broadcast. Out of the 24 scheduled teaching weeks in 2013-14 a total of 18 included a broadcast, whilst in 2014-15 a total of 16 were recorded.

A repository system for students to view on-demand the recordings of the live broadcasted lectures was also offered. We used a cloud-based video distribution service, YouTube, to store and delivery the video. Each video was made private and not for public viewing to ensure that the collected data analytics reports capture in-module viewing. Figure 2 shows the typical playback UI. The comments or post from the iLecture session are made available to the students.

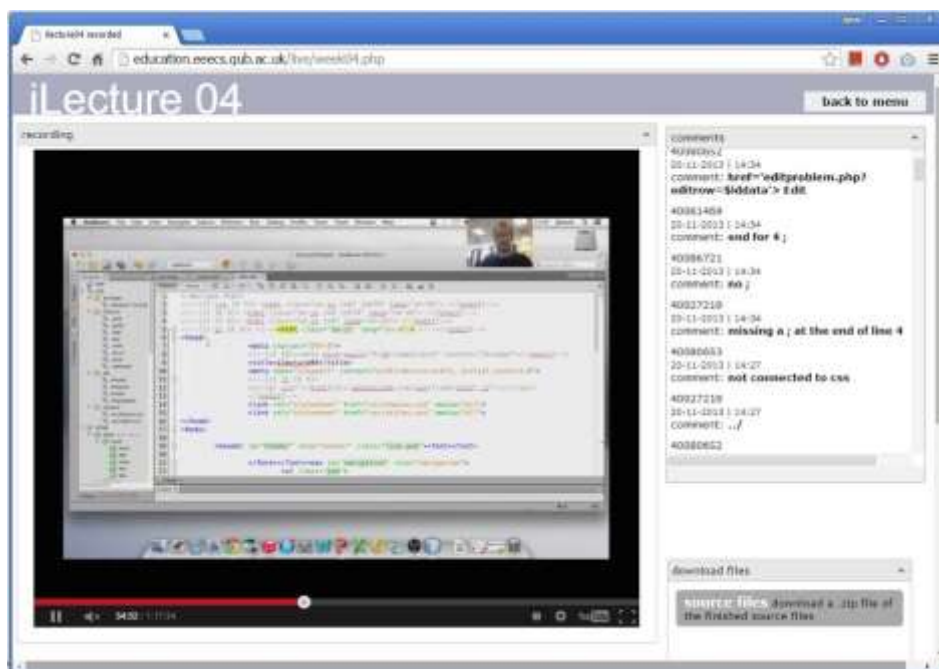


Figure 2. Recorded broadcasting UI

## Method

A total of 378 undergraduate students enrolled on a full-year module on web technology and programming offered during the 2012-13, 2013-14 and 2014-15 academic years. The cohorts take only students who attempted to submit assessments where deemed suited for the study this reduces to 367 students when students who dropped out of the module are discounted.

The module assessment structure uses two individual projects that are submitted at the end of each semester and an end-of-term examination. The project work is worth a total 70% of the module credit, with an equal weighting for each project. The examination accounts for the remaining 30% of course credit.

## Student performance

To explore the potential impact of the iLecture on student performance, a direct comparison of the project assessment marks across the three academic years was undertaken. In the 2012-13 academic year the iLecture system was not implemented, whereas it was used during the 2013-14 and 2014-15 academic years. To measure the impact on student's performance it was decided to use their final classification for the module.

## Student experience

A qualitative approach was used to explore the experiences of the students using the iLecture broadcasting and repository utilities. Two student surveys in 2013-14, one at the end of each teaching semester, and one survey in 2014-15 at the end of the spring semester were conducted. The class of 2013-14 had 97 enrolled students and provided a response frequency for the autumn semester of 76.2% and a spring response frequency of 95.8%. The 2014-15 delivery had 143 students enrolled that resulted in a response frequency of 85.3%.

## Data analysis

Deep analytics can be reviewed using the access logs to explore, visualise and summarise data (Gibson

& Freitas, 2014; Nakayama, Kanazawa & Yamamoto, 2009). This method is used to look for patterns in learner behavior and correlations with the results of the survey. The number of access requests to the web server which related to the date and time of an iLecture were collected and analysed.

The following data was analyzed to determine if the trends and views expressed within the qualitative study mapped onto the usage statistics:

- iLecture student login MySQL tables data
- access logs from the Apache web server.

Student login timestamps were collected and stored when students logged into the iLecture web portal.

Apache web server logs were analysed as a means of working out when students were running code examples. The identity of each individual student's programming activity will only be summarised, rather than explored, in this study. Students were not aware that the access log files would be used to measure participation. The study will focus on 6 iLecture slots in the 2014-15 autumn semester. Each iLecture runs for around 1.5 hours a total of 9 hours, this amounted to over 39k logs entries.

## Results

### Student Performance

A side-by-side comparison of the summative classification results from the 3 academic years 2012-13 to 2014-15 can be seen in Table 1. The upwardly increasing pattern suggests that the intervention of live lecture broadcasted maybe a strong factor in the improvement in student performance. No major changes to the module tutor, content, assessment or learning outcomes for each occurring year. Indicating there are not other factors that could result in the improvement. There is a depression in the number of 1st classifications from 17% in 2013-14 to 13% in 2015-15 but post introduction the percentage of 1st class marks has notable increased. A chi-square test of independence of  $p=0.001$  shows that there is a positive relationship with the introduction of the iLecture system and the improvement of student performance.

Table 1. *Student Performance*

Results	2012-13	%	2013-14	%	2014-15	%	Total	%
70+	8	6	16	17	18	13	42	11
60+	29	22	31	32	45	33	105	29
50+	61	46	32	33	46	33	139	38
40+	21	16	5	5	23	17	49	13
Fail	14	11	12	13	6	4	32	9
Total	133	100	96	100	138	100	367	100

Analysing the data further suggests that mid-range (50+) student's performance exhibits the largest overall improvement. Figure 3 compares all classifications across the three cohorts. The mid-range performing students results show a clear trend of moving from the 2:2 into the 2:1 classification.

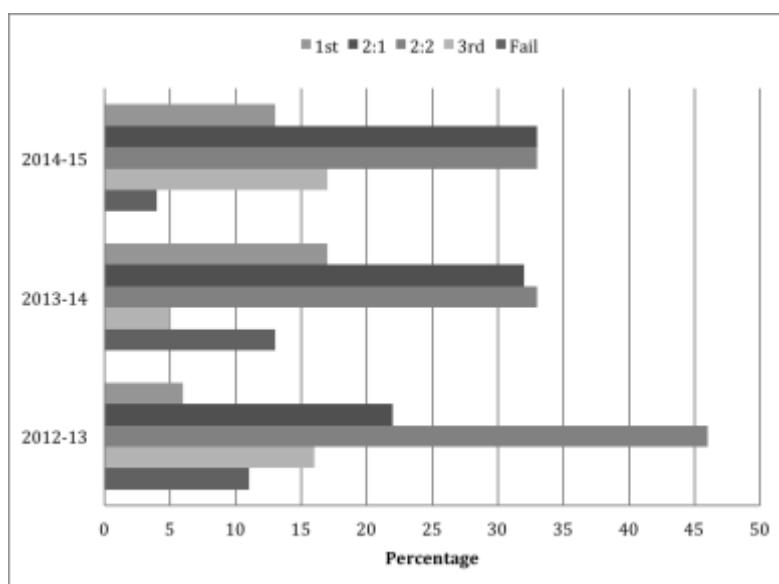


Figure 3. Percentage of students achieving classifications

## Student Surveys

Turning to the student surveys, when asked about their overall learning experience, 'Do the interactive lectures enhance my learning experience?' using a likert scale from 'strongly disagree' (1) to 'strongly agree' (5). The results report a mean score of 4.53,  $SD=0.76$ . Differences cannot be discerned when comparing the cohorts in 2013-14 and 2014-15 since the chi-square statistic is non-significant at  $p=0.51$ . The overall mean score of 4.53 and that 23.18% of the cohort selected either agree and 66.44% selected strongly agree implies that the inclusion of the iLecture system was positively received.

Since the iLecture encouraged students to actively interact with their PCs and copy the code demonstrated by the tutor we wanted to see how the students experienced the pace of the tutor's demonstration. As above, a likert scale from 'too slow' (1) to 'too fast' (5) was presented. The results show the mean at 2.99,  $SD=0.69$  with 62.28% agreeing the speed was fine. This suggests that the majority of students found the speed acceptable. Although 19.72% found it slow or too slow and 17.99% found it fast or too fast. Further analysis shows that there is a significant difference between 2013-14 and 2014-15,  $p<.001$ . The 2014-15 results show 31.15% felt the iLectures speed was too 'slow'. We find this interesting since the content for the live broadcasting changed but not greatly. Further analysis suggests from the open comments of the study points to the students remarking on the 'slowness' of the broadcast.

"slowness of the iLectures, more content could've been discussed"

"a bit slow/too long/drag on needs to be more straight to the point"

Some students directly blamed other students coding errors since the tutor provided direct support and solutions for technical and syntax coding errors during the live broadcast.

"iLectures were useful once John got past the point of answering pointless questions"

"Overall I thought the iLectures were a good concept but I felt they were too prone to being sidetracked by silly questions etc"

"ilectures are brilliant for learning just maybe too much time spent helping people who commenting having a problem. This is fine when watching live but annoying when using recorded."

We might conclude from these results and comments that the students felt the tutor needed to be more aggressive in moving the learning content forward during live broadcasts. Demonstrations of code were slowed down to help the weaker students, which would likely not occur in a lecture as the weak student would not ask, but here the weak student may be a lot more inclined to ask. The design and speed of the broadcast needs to be carefully balanced by making sure weaker students don't get lost with stronger students staying motivated. This still does not explain the significance results between both years.

The students were asked to identify the features of the iLecture system that had a positive affect on their learning. They were presented as:

- Live broadcasting
- Commenting/messaging during the live broadcasts
- Being active rather than passive during the broadcasts – 'learning by doing'
- Recording of live broadcast for playback later.

A summary of the results can be seen in Table 2. Students show positive attitudes to both the recording and playback as well as being active during the lecture. Both returning 83+% averages with no major difference observed between the years. A Pearson's chi-square tested returned  $p > .07$ . The live broadcasting feature does not fair so well, we are surprised at this result since this feature is directly related to the active feature. Although a returned  $p = .05$  does imply that there is some relationship between this feature and the class surveys. The 2014-15 cohort returned a more favourable result at 64.75% for the live broadcast. The new streaming technology in 2014-15 to an in-house solution that decreased the broadcasting lag might have had an impact. Lastly the commenting feature analysis presented  $p < .0001$ , which suggested that differences can be discerned between the different surveys. The 2014-15 group significantly thought positively about the text based commenting function. The authors are slightly perplexed on the reasons since the UI or technology did not change between the years. Overall more questions are likely raised than answered.

Table 2. Summary of student opinions on listed features of system

	Aut_2013-14	Spr_2013-14	Spr_2014-15	Total
Live broadcast - agree	52.7%	49.46%	64.75%	56.75%
Live broadcasting - disagree	47.3%	50.54%	35.25%	43.25%
Commenting - agree	41.89%	45.16%	67.21%	53.63%
Commenting - disagree	58.11%	54.84%	32.79%	46.37%
Learning by doing - agree	87.84%	78.49%	88.52%	85.12%
Learning by doing - disagree	12.16%	21.51%	11.48%	14.88%
Recording - agree	83.78%	81.72%	91.8%	86.51%
Recording - disagree	16.22%	18.28%	13.49%	13.49%

The last section of the survey asked students to identify any issues they had with the iLecture system. The iLecture broadcasting software shows that it was robust with 97.23% having no issues with the video and 95.5% had no issues with the audio, see Table 3. We note there is a high percentage of learners who indicated that they fell behind during the live broadcasting demonstration with no significance among the surveys ( $p = .76$ ). This suggests that nearly a third of the population found it difficult to follow the tutor's demonstration speed. Even though the survey results for the speed of the broadcast show an unequal correlation, only 17.99% found the speed too fast. In the 2014-15 a total of 29.51% indicated they were falling behind, the 70.49% could potentially reason that the speed was acceptable or not quick enough. This denotes that the iLectures were too slow.

Table 3. Summary of student opinion on the iLecture problems

	Aut_2013-14	Spr_2013-14	Spr_2014-15	Total
Video issues - agree	1.35%	4.3%	2.46%	2.77%
Video issues - disagree	98.65%	95.7%	97.54%	97.23%
Audio issues - agree	1.35%	10.75%	1.64%	4.5%
Audio issues - disagree	98.65%	89.25%	98.36%	95.5%
Fall behind instruction - agree	33.78%	29.03%	29.51%	30.45%
Fall behind instruction - disagree	66.22%	70.97%	70.49%	69.55%

## Data Analysis results

The nature of this student survey answers are self-reporting which can introduce conflicting results. Data analytics can be used to support or oppose the reported findings. Figure 4 shows a summary of attendance rates of iLectures in 2014-15. The data shows that attendance rates dropped significantly in semester 2. We assume that the broadcast being timetabled for Friday afternoon had a severe impact on student participation.

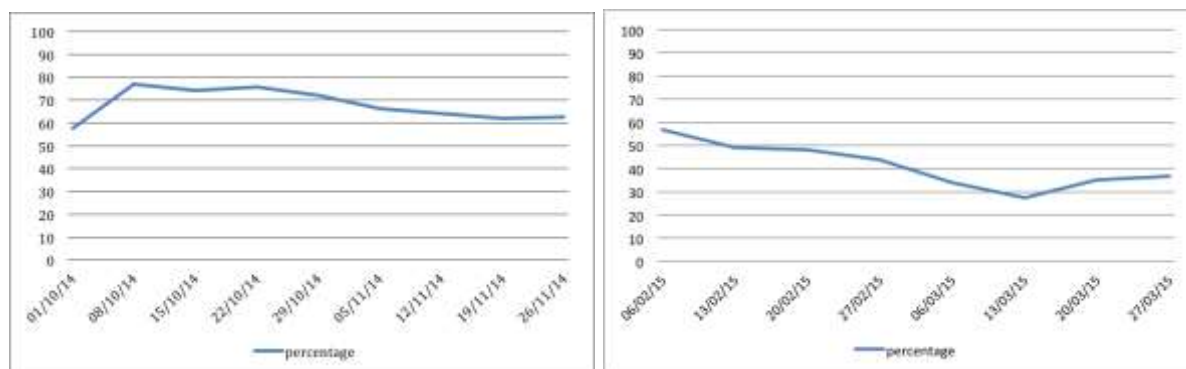


Figure 4: Attendance rates of iLecture broadcasting

Figure 5 summarises the student code-based execution activities during the 2014-15 autumn semester. Note that the data collected from the access logs only covers the last 6 weeks of the spring term. This is due to students using a client-side technology that didn't create access logs in the earlier part of the module. The chart provides a simplistic illustration of student activity over six points (the last 6 iLecture broadcasts). It shows a gradual incline from around 0, students had little log data since they were working on client-based technologies that were not stored by the Apache server. The content cover was general basics of introductory PHP programming. We would have not accepted a lot of log entries so is deemed normal. Interestingly there is a large spike at week 8's broadcast. This is due to student increased student attendance since figure 4 shows the attendance is at similar levels from previous weeks. It can be posed that the tutor's demonstration content created this peak during the broadcast. We reviewed the recorded live broadcast and found that the content covered material that needed to be understood in order to complete project 01.

This would imply that learning content that will directly relate to the module assessment will increase student participation.

Learner analytic data created from this study has the potential to improve tutor confidence and teaching strategy. By evaluating this summary and identifying the high levels of activity from the cohort during the live lecture broadcast. The tutor will gain valuable insight into their students' learning habits and behaviours.



Reviewing Figure 5, learner analytics does report that students were actively engaged during each iLecture. It shows good levels of student activity in the form of 'doing rather than listening'.

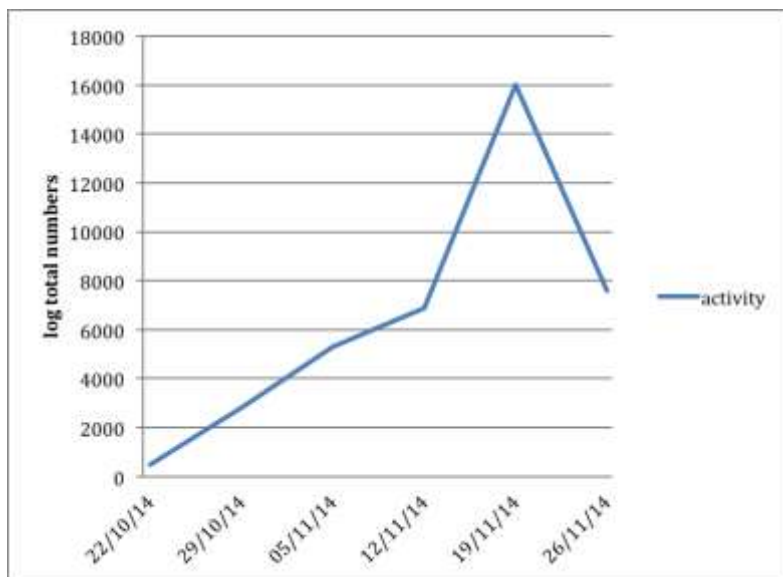
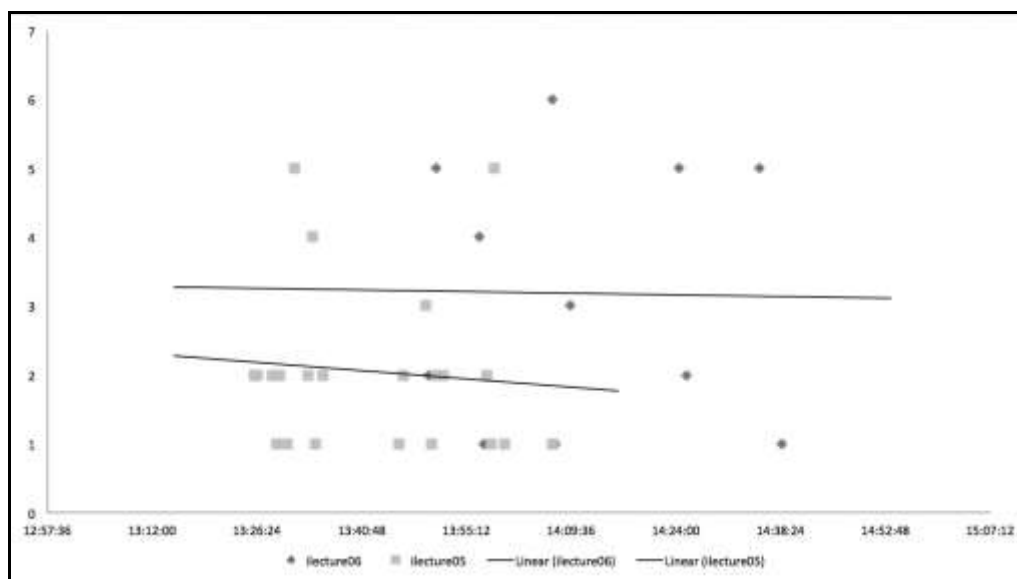


Figure 5. Student code execution logs over 6 iLectures

Using the log entries we can go deeper into an individual student activities to produce a summary of each iLecture from the single student perspective. Figure 6 shows one random student over four weeks. Using the very precise summary we can see that the student's activity was steady throughout iLectures 05, 06 and 08. In iLecture 07 the activity trend descends. Further sample data from the logs would need to be analysed to prove if there was a relationship with the iLecture content and delivery. Or did the student fail to keep up with the live demonstration instructions.



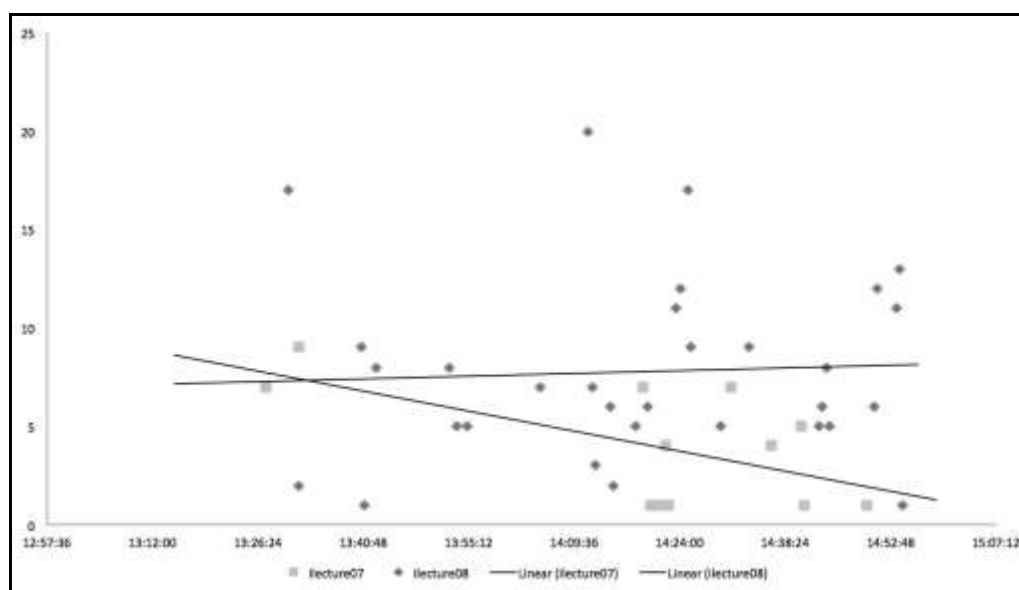


Figure 6. Individual activity logs of one student

## Conclusion

This study on the implementation of a web-based broadcasting lecturing system to enhance the learning experience of student enrolled on a web technology and programming module demonstrates positive results. We showed that the iLecture system encourages students to become more active during lectures. Which, in turn, is likely to lead to improved learner performance. The views from the student were very positive. By comparing each year's summative results, they indicated an improved student performance.

Combining student surveys with an analysis of usage provides a means of analysing and understanding the behaviour of students in large class sizes. The analytics may also be used for tutor peer reflection and improvement.

The live broadcasting speed is a major design subject, especially if students are actively replicating the tutor's instruction. The pace of the live interactive lecture is very important. If students struggle to keep-up with the instruction it will most likely increase the interruption rate. Future work will look at understanding how best to support students of both high and low ability, pedagogy content design, other disciplines, pace of delivery etc. With further analysis of the data analytics it should be possible to answer these questions and produce a 'best practices' matrix regarding live broadcasting lectures for Higher Education.

## References

- Anderson, R., Anderson, R., Davis, K. M., Linnell, N., Prince, C., & Razmov, V. (2007). Supporting active learning and example based instruction with classroom technology. *ACM SIGCSE Bulletin*, 39(1), 69-73. doi:10.1145/1227504.1227338
- Atkins, M., Brown, G. A., & Brown, G. (2002). *Effective teaching in higher education*. Routledge.
- Biggs, J., & Tang, C. (2007). *Teaching for Quality Learning at University*. (3rd Edition). New York, NY: McGraw-Hill.
- Bonk, C. J., & Zhang, K. (2009). *Empowering online learning: 100+ activities for reading, reflecting,*

*displaying, and doing*. John Wiley & Sons.

Boyle, T. (1997). *Design for Multimedia Learning*. London, Prentice Hall.

Curzon, L. B. (2003). *Teaching in Further Education*. (6<sup>th</sup> Edition). London, England: Continuum.

Dey, E. L., Burn, H. E., & Gerdes, D. (2009). Bringing the classroom to the web: Effects of using new technologies to capture and deliver lectures. *Research in Higher Education*, 50(4), 377-393. doi: 10.1007/s10734-014-9786-1

Evans, C. (2008). The effectiveness of m-learning in the form of podcast revision lectures in higher education. *Computers & education*, 50(2), 491-498. doi: 10.1016/j.compedu.2007.09.016

Exley, K., & Dennick, R. (2004). *Giving a Lecture from Presenting to Teaching*. New York, NY: RoutledgeFalmer.

Exley, K., & Dennick, R. (2004). *Giving a lecture: from presenting to teaching*. Routledge.

Fernandez, V., Simo, P., & Sallan, J. M. (2009). Podcasting: A new technological tool to facilitate good practice in higher education. *Computers & Education*, 53(2), 385-392. doi: 10.1016/j.compedu.2009.02.014

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415. doi: 10.1073/pnas.1319030111

Fry, H., Ketteridge, S., & Marshall, S. (2000). *A Handbook for Teaching and Learning in Higher Education*. London, England: Logan Page.

Georgiou, H., & Sharma, M. D. (2015). Does using active learning in thermodynamics lectures improve students' conceptual understanding and learning experiences?. *European Journal of Physics*, 36(1), 015020. doi:10.1088/0143-0807/36/1/015020

Gibson, D., & de Freitas, S. (2014). Exploratory learning analytics methods from three case studies. In B. Hegarty, J. McDonald, & S.-K. Loke (Eds.), *Rhetoric and Reality: Critical perspectives on educational technology. Proceedings ascilite Dunedin 2014* (pp. 383-388).

Gill, M., & Greenhow, M. (2010). Building Student Confidence in Mathematics and Numeracy. In P. Hartley, J., Hilsdon, C., Sinfield, C., Keenan, & Verity, M. (Eds.), *Learning Development in Higher Education* (pp. 129-142). Basingstoke, England: Palgrave Macmillan.

Hürst, W., & Waizenegger, W. (2006). An overview of different approaches for lecture casting. In *proceedings of IADIS international conference mobile learning, Dublin, Ireland*.

Jaki, T. (2009). Recording lectures as a service in a service course. *Journal of Statistics Education*, 17(3), 1-13.

McConnell, J. J. (1996). Active learning and its use in computer science. *ACM SIGCSE Bulletin*, 28(SI), 52-54. doi: 10.1145/1013718.237526

Milne, I., & Rowe G. (2002). Difficulties in Learning and Teaching Programming - Views of Students and Tutors. *Education and Information Technologies*, 7, 55-66. doi: 10.1023/A:1015362608943

Nakayama, M., Kanazawa, H., & Yamamoto, H. (2009). Detecting incomplete learners in a blended

- learning environment among Japanese university students. *International Journal of Emerging Technologies in Learning (IJET)*, 4(1). doi: 10.3991/ijet.v4i1.659.
- Odhabi, H., & Nicks-McCaleb, L. (2011). Video recording lectures: Student and professor perspectives. *British Journal of Educational Technology*, 42(2), 327-336. doi: 10.1111/j.1467-8535.2009.01011.x.
- Pale, P., Miletic, I., Kostanjcar, Z., Pandzic, H., & Jeren, B. (2007). Pyramidia-an integrative e-learning tool. In *EUROCON, 2007. The International Conference on &# 34; Computer as a Tool&# 34;* (pp. 2459-2464). IEEE. doi: 10.1109/EURCON.2007.4400419
- Race, P. (2014). *The lecturer's toolkit: a practical guide to assessment, learning and teaching*. Routledge.
- Reece, I., & Walker, S. (2003). *Teaching, Training and Learning a Practical Guide*. (5<sup>th</sup> Edition). Sunderland, England: Business Education Publishers Limited.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13, 137-172. doi: 10.1076/csed.13.2.137.14200
- Sjöstedt, R. (2015). Assessing a Broad Teaching Approach: The Impact of Combining Active Learning Methods on Student Performance in Undergraduate Peace and Conflict Studies. *Journal of Political Science Education*, 11(2), 204-220. doi: 10.1080/15512169.2015.1030502
- Vihavainen, A., Paksula, M., & Luukkainen, M. (2011, March). Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 93-98). ACM. doi: 10.1145/1953163.1953196
- Wolff, M., Wagner, M. J., Poznanski, S., Schiller, J., & Santen, S. (2015). Not another boring lecture: Engaging learners with active learning techniques, *The Journal of Emergency Medicine*, 48(1), 85–93. doi:10.1016/j.jemermed.2014.09.010
- Zhu, E., & Bergom, I. (2010). Lecture capture: A guide for effective use. *University of Michigan CRLT Occasional Papers* (27).