https://studentsuccessjournal.org/
Volume **12** (1) 2021 https://doi.org/10.5204/ssj.1668

Promoting Engagement in a CS1 Course with Assessment for Learning. *A Practice Report*

Simon Liénardy, Benoit Donnet and **Laurent Leduc**
Université de Liège, Belgium

Abstract

This practice report discusses the evolution of a CS1 Course taught at the University of Liège, Belgium. Over the last seven years several teaching activities have been thought to complement traditional theoretical courses and exercise sessions in order to promote students' engagement. The result is aligned with (i) the principles of assessment for learning, which consists in leveraging the assessment to improve the students learning, and (ii) the concept of blended learning. This report describes the difficulties the students faced and what we implemented to assist our course evolution. We also present and discuss results showing that, despite a high drop-out rate, we managed to engage students to work on a regular basis and, in some cases, raise their performance levels.

***Keywords**:* Assessment for learning; engagement; CS1; blended learning.

Background

Teaching *Introduction to Programming* (i.e., CS1) is known to be a difficult task for many students and has been the topic of a large number of research studies (Luxton-Reilly et al. 2018, Medeiros et al., 2019). In recent years, various studies have demonstrated that, often, students following a CS1 class encounter difficulty in understanding how a program works (Schröter et al., 2017), how to design an efficient and elegant program where conditionals and loops have proven to be particularly problematic (Dale, 2006), (Cherenkova et al., 2014), problem solving and mathematical ability (Medeiros et al., 2019), and in checking whether a program works correctly (Ben-David Kolikant & Mussai, 2008). These are particular examples of difficulties faced by students we also observed in our CS1 class. The course we teach targets first year computer science students and introduces basic principles of computer science and, in particular, the basic principles of programming. The course is offered at the Montefiore Institute, which is the department of Electrical Engineering and Computer Science of the Faculty of Applied Sciences of the University of Liège, a major public university of the French Community in Belgium.

For a long time, the failure rate and withdrawal ratio of our students have been high, as it is the case with other CS1 courses (Beaubouef & Mason, 2005; Watson and Li, 2014). Moreover, the failure rate in these particular courses reaches around 70% for first-year students at universities in Belgium (ARES, n.d.). This figure can be partially explained by the open access policy for higher education in Belgium. This has meant that some students come to university without important background skills for particular courses of study. However, this situation should not be taken as a pretext for inaction but rather as a call for the improvement of courses and teaching.

Literature in educational psychology has highlighted many predictors of student success in higher education (Dupont et al., 2015). Among the various parameters favouring it, some can be influenced by the teacher practices, namely those that lead to more student engagement (Lizzio et al., 2002).

A few years ago, we decided to make our CS1 course change from a traditional course (i.e., *ex cathedra* theoretical lessons with exercises sessions) towards a new approach that would promote engagement, potentially be more motivating for first year students, and that would provide them with more feedback. With this in mind, the *Assessment for Learning* (AfL) approach seemed to us a promising way to achieve our goals. Sambell et al. (2013) describes AfL as:

> [A]n integrated approach to teaching, assessment and supporting student learning. Our view of assessment is broad. It includes summative assessment activities but also assessment which plays a vital role in improving students' progress and attainments and is embedded in teaching (p. 147)

This report discusses that evolution of our CS1 course over the last seven years. We believe that experience gained with this course evolution is quite general and could be easily transferred by any educational team eager to adopt AfL for their course.

Evolution of the CS1 Course

Our CS1 course is provided during the first semester at the University of Liège (between mid-September until mid-December). During the All-Saints week, mid-term exams (Mathematics, Physics, CS1, and Foreign Language) are organised for first year students. After two revision weeks in December, the final exam is held in January.

The CS1 course has been offered students since 2013 and, originally consisted of traditional theoretical lectures, exercises (i.e., programming tasks on paper), and lab sessions (i.e., programming tasks in front of computers). Regarding the assessment, we followed Faculty guidelines by organising the mid-term and the final exam. In order to ensure the students learn the theory (i.e., the rules of the programming language that must be known to ensure the course understanding as the semester goes on), five short multiple choice questions (MCQ) tests (Cicirello, 2009) were organised during the semester. In 2015, due to the low passing rate in the Computer Science section, remedial classes were introduced and made mandatory for students failing at the mid-term exams (not only for the CS1 course but also in Mathematics, Physics, and Foreign Language). The core change took place in 2016 when we introduced a *Programming Challenges Activity* (PCA), i.e., a teaching activity spread over the whole semester and consisting in submitting small pieces of code (called *Challenges*) on a platform, called CAFÉ (Correction Automatique et Feedback des Étudiants), that immediately corrects them and provides students with both feedback (i.e., information about what is wrong with their solution) and, starting 2018, feedforward (i.e., what can be done to improve the solution) (Liénardy et al., 2020). The students can then use this feedback and feedforward to make a new submission that eventually may lead to a better performance (Boud 2000). Students can submit again up to two times, hence totalling three submissions. Karavirta et al. (2006) have shown that students submitting a lot of time pieces of code for online assessment without necessarily getting good grades are inefficient in their work. As a conclusion, Karavirta et al. suggested to limit the number of resubmissions in order to "guide [students'] learning process" (p. 238). This is exactly what has been done with CAFÉ. We decided to limit the number of submissions to three to avoid 'trial and error' process that is in contradiction with our programming methodology (Liénardy et al., 2020). Students have three days to complete a Challenge. Moreover, during the semester, students are allowed to skip one of the Challenges (except the Challenge 0 designed to get students familiar with the submission process on CAFÉ), what we call 'playing one's Trump Card'. This means that, when the Trump Card is played by a student, the Challenge does not account in the student's final mark. It is enough not to submit any response to a Challenge to play the Trump Card. This possibility was set up for two reasons: first, to avoid student excuses when not submitting their Challenge (Brauer, 2011), second, to increase students' perception of controllability, inducing higher motivation (Viau, 2009).

As such, the introduction of the PCA marked the beginning of the course transformation into blended learning (Bonk & Graham, 2006). In 2018, the subject covered in the last Challenge was extended to the last chapter of the course because we observed the students had struggled to master it.

Assessment for Learning

Since the introduction of the PCA, all teaching activities have complemented each other to align with the six principles of Assessment for Learning (AfL) presented by Sambell et al. (2013):

1. The PCA and both mid-term and final exams consist of solving genuine programming tasks. These do not only assess the final product (i.e., the program) but also the use of a proper programming methodology that led to the formation of the program. This refers to the AfL principle of "authentic assessment" (Sambell et al., 2013, p. 6).
2. CAFÉ offers students the possibility to have up to two free submissions (they receive feedback about their performance without affecting their grades). Moreover, prior to MCQs, students have the opportunity to train themselves with mock MCQs, available on the course blackboard. This ensures to "balance summative and formative assessment" (Sambell et al., 2013, p. 6).
3. Students may submit an answer up to three times during each Challenge (Karavirta et al. 2006) and receive feedback after each submission. Mock MCQs provide students with immediate feedback. By doing so, we "create opportunities for practice and rehearsal" in "low-stakes teaching activities" (Sambell et al., 2013, p. 6). Even though both MCQs and PCA are graded, they do not account for a large amount in the final mark (10% for the MCQs and 10% for the PCA).
4. The feedback provided by CAFÉ after each submission, the correction of the mid-term exam**,** and the mandatory remedial courses provide students with "formal feedback to improve learning" (Sambell et al., 2013, p. 6). Formal feedback encompasses teacher's comments on student's work as well as and "self- and peer review and reflection" (Sambell et al., 2013, p. 6).
5. The PCA introduction has enabled us to also change the way we organise the exercise and lab sessions. The students work in small groups in the classroom and can discuss with each other and with the pedagogical team gaining "opportunities for informal feedback" (Sambell et al., 2013, p. 6).
6. As the course focuses on a programming methodology, it also covers the good practices of a programmer: how to test one's code, how to efficiently search for help in the proper resources (documentation, Internet, etc.) in order to turn students into "effective lifelong learners" (Sambell et al., 2013, p.7).

We anticipate these six principles will contribute to increasing students' engagement.

Measuring the Impact

Evolution of the Course Population

Table 1 presents the different cohorts that took the course between 2013–2014 and 2019–2020. The total number of students rose from 2013 to 2019 (a growth rate of 41%). The proportions of students that come for the first time to the university (labelled as *fresh people*) are 76.1%, repeaters 16.3%, and transfer students (i.e., those who reoriented from another curriculum).

Table 1

Populations of the CS1 Course from Academic Years 2013–2014 to 2019–2020

Academic Year	Total	Fresh people		Repeater		Transfer students		Course Results	
		#	%	#	%	#	%	SR	FR
2013–2014	58	43	74	4	7	11	19	25.9	74.1
2014–2015	58	42	72	5	9	11	19	12.1	87.9
2015–2016	52	41	79	1	2	10	19	17.3	82.7
2016–2017	54	42	78	3	6	9	17	38.9	61.01
2017–2018	72	53	74	9	13	10	14	25.0	75.0
2018–2019	76	64	84	6	8	6	8	23.7	76.3
2019–2020	82	59	72	8	10	15	18	19.5	80.5

Notes. SR and FR stand respectively for Success Rate and Failure Rate (the failure rate also includes students that did not take the exam) to the course, in January.

Data Sources

Numerous data sources are considered, i.e., data directly retrieved from CAFÉ, the course Blackboard page, and the results of the assessments. Except for teaching activities results, the analysis is focused on data about students' engagement. First, the platform on which CAFÉ is run enables us to retrieve data about students' submissions, namely the number of submissions per Challenge and the students' results for each Challenge. Second, the Challenge subjects can be downloaded from the Blackboard platform of the course. The Blackboard usage statistics enable us to collect data about mock MCQs. Finally, data about participation in all the assessments are used: there are five MCQs, five Challenges, the mid-term exam, and the final exam

Results

Student's Engagement

Figure 1 presents the MCQs participation rate from 2013 to 2019. Since the MCQs take place at the beginning of the session, these figures are a proxy of the course attendance. The first MCQs have a participation rate higher than 80%. It decreases a bit for the third MCQs that take place shortly after the mid-term exam but before the results are made available to the students. Once they are, the participation rate drops for MCQ4 and the participation rate decrease continues for MCQ5. That rising number of absences through the semester is indicative of abandonment since the MCQs take place at the beginning of exercises sessions. In Figure 1 the participation rates are shown as decreasing over the years (except in 2016–2017), suggesting that, while the student numbers increased during that period (as can be seen in Table 1), the absenteeism rate increased as well. This may be partially explained by the open access policy to higher education in Belgium: students may enter the CS curriculum lacking some fundamental skills (e.g., in maths) and are discouraged (e.g., by maths and physics courses).

Figure 1 shows also the participation to mock MCQs (at the right of the Figure) in 2018 and 2019. In 2018, the training rate is 10% below the participation rate for the MCQs 1 and 2. It drops to 20% below the participation rate for the next MCQs. In 2019–2020, the participation rate rose while the training rate decreased, leading to a difference of 20% or more between the training and the corresponding MCQ.

Figure 1

Participation in the MCQs from Academic year 2013-2014 to 2019-2020

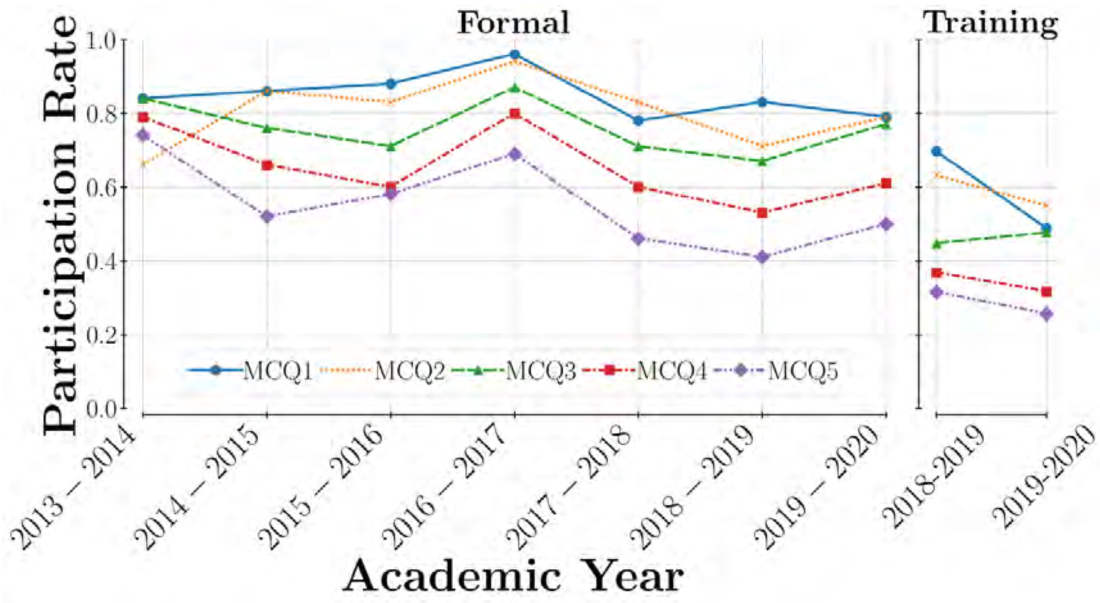
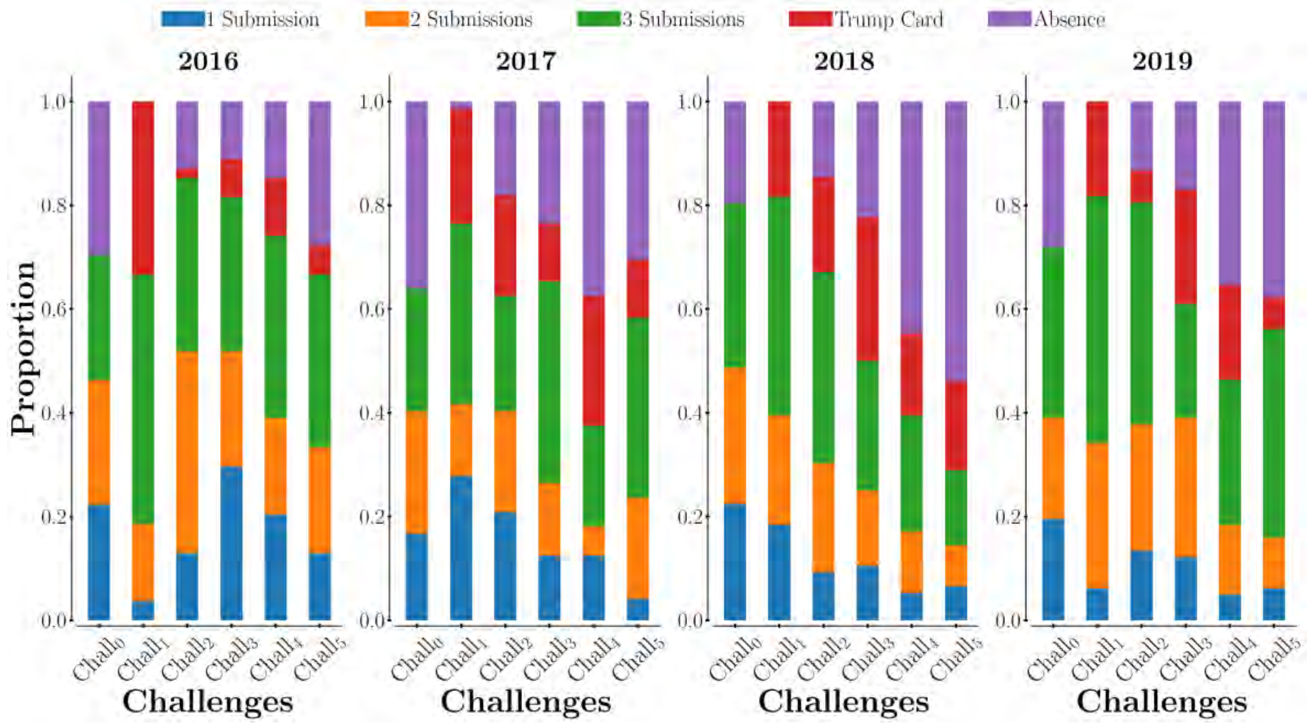


Figure 2

Participation to the PCA, on a Per Challenge Basis, from Academic year 2016–2017 to 2019–2020



Notes. In 2017, the absence for the Challenge 1 refers to a student that joined the curriculum later in the semester.

With respect to Challenges participation, Figure 2 shows the proportion of the number of submissions, as well as the proportion of students who did not submit the Challenge either because they were absent or they played their Trump Card. Students who take the Challenges mainly submit three times and may benefit from the feedback. This is exactly what CAFÉ was made for, with AfL in mind, i.e., to offer "opportunities for practice and rehearsal" (AfL, 3rd principle). On the other hand, the number of absentees grows as the semester goes and there is often a big rise between the Challenge 3 and 4, i.e., when the results of the mid-term exams are published to the students.

Participation figures in the mid-term and final exam are presented in Table 2. This table shows that the participation in the mid-term exam is fairly constant and always above 80% except in 2015-2016 (77%). There are few students that sign the presence sheet without taking the exam at mid-term and the proportions of absences vary between 9% and 15%, except in 2015-2016 (21%). The introduction of the PCA does not seem to have affected the mid-term exam participation. On the other hand, the final exam participation is decreasing over time (except in 2016-2017 that is an outlier, as for Challenge and MCQs participation). The introduction of the PCA does not seem to have reduced that trend. On the contrary, the mean participation rate is 82.3% before the PCA and 75% after its introduction. Note that this drop in the participation to the final exam is actually counter-balanced with an increase in the success rate (see next section).

Table 2

Participation in the Mid-Term and Final Exam from Academic year 2013–2014 to 2019–2020

Academic Year	Mid-Term Exam				Final Exam			
	Part.	Sign.	Abs.	Exc.	Part.	Sign.	Abs.	Exc.
2013-2014	91%	0%	9%	0%	87%	4%	9%	0%
2014-2015	88%	0%	10%	2%	83%	5%	12%	0%
2015-2016	77%	0%	21%	2%	77%	6%	15%	2%
2016-2017	89%	2%	9%	0%	89%	6%	6%	0%
2017-2018	86%	0%	13%	1%	78%	10%	10%	3%
2018-2019	80%	0%	15%	5%	61%	21%	18%	0%
2019-2020	86%	0%	10%	4%	73%	16%	11%	0%

Notes. “Part.” is the proportion of students taking the exam; “Sign.” is the proportion of students that are present without taking the exam (they sign the presence sheet); “Abs.” is the proportion of absentees and “Exc” stands for “Excused” (i.e., students who cannot be present for a good reason, generally a medical one).

Students’ Performance

Table 3 shows the link between participation to Challenges and the students’ performance from Academic Year 2013-2014 to 2019-2020. For each year the PCA has been organised, the table shows the cumulative number of students having submitted Challenges and the mean number of completed Challenges per student (i.e., “Challenges Participation” columns). The table also shows the cumulative number of Challenge submissions over the whole PCA, the cumulative average number of submissions per student over the whole PCA, and the mean number of submissions per individual participation (i.e., “PCA Participation” columns). Table 3 also provides data about students’ performance: the success and failure rate (resp. SR and FR) in January (students can retake the exam in June and August). Focusing on the students that take the exam enabled us to evaluate more precisely the effect of the PCA introduction on the success rate. The success rate dropped between 2013 to 2016, before the introduction of the PCA. Starting from the introduction of the PCA, in 2016-2017, the success rate increases. From 2015-2016 to 2016-2017, it almost doubled although the year 2016-2017 seems to be an outlier in term of students’ participation (4.44/5 challenges per student) and success (43.8% actually taking the exam). The shift of the success rate may be explained both by the fact that students can work on a regular basis thanks to the Challenge and because the PCA grades take into account in their final mark, allowing them to increase it in a low-stakes activity (AfL, 3rd principle).

Table 3

Link Between Students Participation and Performance from Academic Years 2013–2014 to 2019–2020

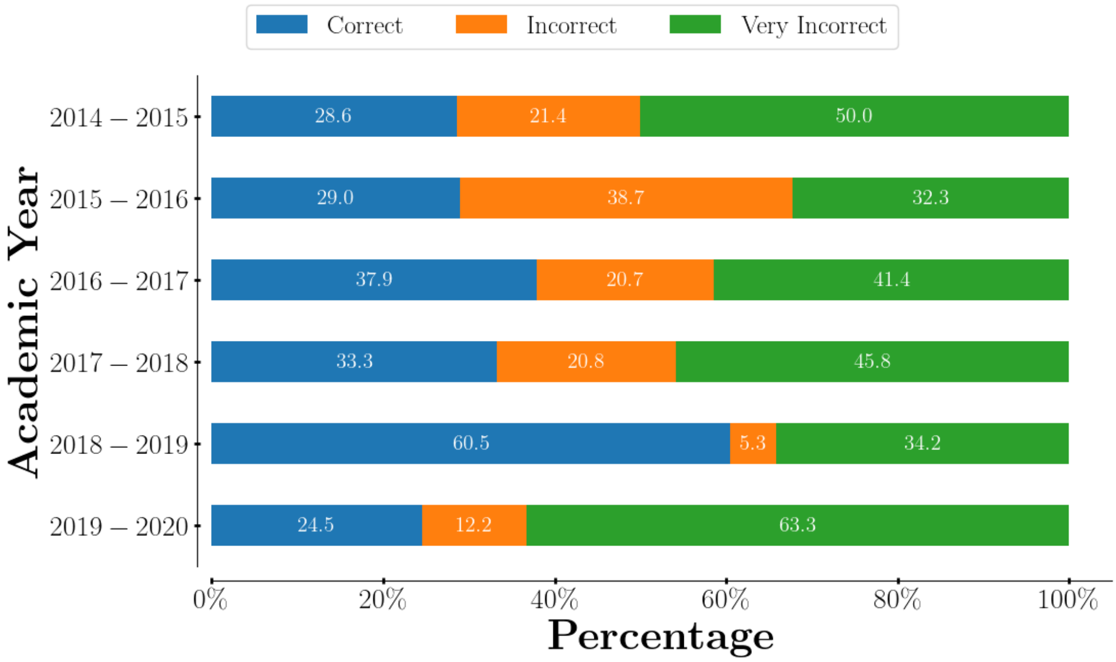
Academic Year	# Students	Challenges Participation		PCA Participation			Course Performance			
							Total		Actually taking	
		#	Per Stu.	#	Per Stu.	Per Part.	SR	FR	SR	FR
2013–2014	58	-	-	-	-	-	25.9	74.1	33.3	66.7
2014–2015	58	-	-	-	-	-	12.1	87.9	14.6	85.4
2015–2016	52	-	-	-	-	-	17.3	82.7	22.5	77.5
2016–2017	54	240	4.44	535	9.91	2.23	38.9	61.1	43.8	56.3
2017–2018	72	214	2.97	478	6.64	2.23	25.0	75.0	32.1	67.9
2018–2019	76	202	2.66	473	6.22	2.34	23.7	76.3	39.1	60.9
2019–2020	82	267	3.21	647	7.80	2.42	19.5	80.5	26.7	73.3

Notes. "#" refers to cumulative values. "Per Stu." refers to the average per student. "Per Part." is the mean number of submissions per Challenge participation. SR and FR stand respectively for Success Rate and Failure Rate. Total figures include students that did not take the exam while "Actually taking" figures only include the students that took the Final Exam (in January). There is no data from 2013 to 2015 regarding the Challenges and PCA since CAFÉ was not yet implemented.

Another way to assess the effect of the PCA is to look at the quality of code written by the students at the Final Exam, in January. This is shown in Figure 3. The figure shows the proportion of codes that are Correct (i.e., the code has the intended behaviour), Incorrect (i.e., the student committed a minor mistake), or Very incorrect (i.e., the code presents syntax errors or does not make any sense or contains a major mistake). 2018-2019 and 2019-2020 are outliers: the exam question may have been too easy in 2018-2019 while it may have been too difficult in 2019-2020, leading to respectively more Correct (60.5%) and Very Incorrect codes (63.3%). Nevertheless, the proportion of code labelled as Correct increases from 28.6% (2014-2015) and 29.0% (2015-2016) without the PCA to 37.9% (2016-2017) and 33.3% (2017-2018) after the introduction of the PCA. Consequently, those who seriously engage in the PCA (and thus have practiced and take advantage of feedback and feedforward throughout the semester) are more likely to write a correct piece of code at the end of the course.

Figure 3

Final Exam Code Quality from Academic Years 2014–2015 to 2019–2020



Note. "Correct" refers to codes that are (nearly) correct. "Incorrect" refers to codes that contain minor mistakes and "Very Incorrect" refers to codes either that do not make any sense, presents syntax errors, infinite loops or out-of-bound memory accesses.

Discussion and Lessons Learned

The data presented in this report shows that after maturation of the CS1 course and alignment with AfL principles has increased student engagement. Students have received earlier feedback and feedforward to improve their performance. However, the data has shown that the students do not take all the offered opportunities to train themselves as far as, for example, the mocks MCQs are concerned.

Besides, the mean success rate (in January – students can retake the exam in June and August) has risen from 18.4% between 2013 and 2015 to 26.8% (+ 45%) between 2016 and 2019 (See Table 1). It is encouraging to think that these figures can partially be attributed to the course evolution, as suggested in Table 3.

The number of students exiting during the semester is still high, in particular after the mid-term exam results are reported. This was one of the reasons to introduce AfL principles in the course, especially feedback that may "Encourage positive motivational beliefs and self-esteem" (Nicol & Macfarlane-Dick, 2006). Those students dropping out cannot be totally eliminated, as their actions may be the result of the open-access policy to higher education in Belgium. From our perspective it is unfortunate to lose such a large portion of our students. We hope that the CS1 course helps students understand the expectations and requirements of a university curriculum, in terms of work and commitment and that they go on to succeed

References

ARES. (n.d.). Indicateurs de l'enseignement supérieur. https://www.ares-ac.be/en/statistiques/indicateurs

Beaubouef, T., & Mason, J. (2005). Why the high attrition rate for computer science students: some thoughts and observations. *ACM SIGCSE Bulletin*, *37*(2), 103-106. https://doi.org/10.1145/1083431.1083474

Ben-David Kolikant, Y., & Mussai, M. (2008). "So my program doesn't run!" Definition, origins, and practical expressions of students'(mis) conceptions of correctness. *Computer Science Education*, *18*(2), 135–151. https://doi.org/10.1080/08993400802156400

Bonk, C. J., & Graham, C. R. (2006). *Handbook of blended learning: Global perspectives, local designs.* John Wiley & Sons. https://doi.org/10.5465/amle.2008.31413871

Boud, D. (2000). Sustainable assessment: Rethinking assessment for the learning society. *Studies in Continuing Education*, *22*(2), 151-167. https://doi.org/10.1080/713695728

Brauer, M. (2011). *Enseigner à l'université: conseils pratiques, astuces, méthodes pédagogiques.* Armand Colin.

Cherenkova, Y., Zingaro, D., & Petersen, A. (2014). Identifying challenging CS1 concepts in a large problem dataset. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 695–700. https://doi.org/10.1145/2538862.2538966

Cicirello, V. A. (2009). On the role and effectiveness of pop quizzes in CS1. *ACM SIGCSE Bulletin*, *41*(1), 286–290. https://doi.org/10.1145/1539024.1508971

Dale, N. B. (2006). Most difficult topics in CS1: Results of an online survey of educators. *ACM SIGCSE Bulletin*, 38(2), 49–53.https://doi.org/10.1145/1138403.1138432

Dupont, S., De Clercq, M., & Galand, B. (2015). Les prédicteurs de la réussite dans l'enseignement supérieur. Revue critique de la littérature en psychologie de l'éducation. *Revue Française de Pédagogie*, *191*(2), 105–136. https://doi.org/10.4000/rfp.4770

Karavirta, V. & Korhonen, A., & Malmi, L. (2006). On the use of resubmissions in automatic assessment systems. *Computer Science Education*, *16*(3), 229–240. https://doi.org/10.1080/08993400600912426

Liénardy, S., Leduc, L., Verpoorten, D., & Donnet, B. (2020). CAFÉ: Automatic correction and feedback of programming challenges for a CS1 Course. *Proceedings of the Twenty-Second Australasian Computing Education Conference*, 95–104. https://doi.org/10.1145/3373165.3373176

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L. M., Scott, M. J., Sheard, J., & Szabo, C. (2018). Introductory programming: A systematic literature review. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 55–106. https://doi.org/10.1145/3293881.3295779

Lizzio, A., Wilson, K., & Simons, R. (2002). University students' perceptions of the learning environment and academic outcomes: Implications for theory and practice. *Studies in Higher Education*, *27*(1), 27-52. https://doi.org/10.1080/03075070120099359

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, *62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, *31*(2), 199-218. https://doi.org/10.1080/03075070600572090

Sambell, K., McDowell, L., & Montgomery, C. (2013). *Assessment for learning in higher education*. Routledge. https://doi.org/10.4324/9780203818268

Schröter, I., Krüger, J., Siegmund, J., & Leich, T. (2017). Comprehending studies on program comprehension. *Proceedings of the 25th IEEE/ACM International Conference on Program Comprehension (ICPC)*, 308–311. https://doi.org/10.1109/ICPC.2017.9

Viau, R. (2009). *La motivation en contexte scolaire.* De Boeck.

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. *Proc. Conference on Innovation & Technology in Computer Science (ITiCSE)*, 39-44. https://doi.org/10.1145/2591708.2591749

Please cite this article as:

Liénardy, S., Donnet, B., & Leduc, L. (2021). Promoting engagement in a CS1 course with assessment for learning. A practice report. *Student Success*, *12*(1), 102-111. https://doi.org/10.5204/ssj.1668

This practice report has been accepted for publication in *Student Success.* Please see the Editorial Policies under the 'About' section of the Journal website for further information.