34th Australasian Association for Engineering Education Conference
3 - 6 December 2023 – Gold Coast, QLD

Learning to program – a tale of two cohorts

Melanie Fleming[a]; Peter O'Shea[a], Paul Vrbik[a], Brae Webb[a], Greg Birkett[a]
The University of Queensland[a],
Corresponding Author Email: melanie.fleming@uq.edu.au

ABSTRACT

CONTEXT

The complexities of learning to program are well known (Gomes & Mendes, 2007; Muresano et al., 2010). Educators often provide additional supports or resources to students to assist their learning (Mehmood et al., 2020). At the University of Queensland, as part of a curriculum review process implemented in 2021, all Bachelor of Engineering students now complete an introductory programming course (DOGS101). The course complements an existing programming course targeted to those students more likely to complete a major in software, electrical or mechatronics engineering (CATS101).

PURPOSE OR GOAL

This study explored how different cohorts of students interact with the resources provided to assist them to learn to program. Students can access the First Year Programming Learning Centre (or FYPLC, a new initiative in 2022) for tutor support; ask questions in an online discussion forum (EdStem); and receive support prior to assessment submission through an automated feedback process (style, testing etc).

APPROACH OR METHODOLOGY/METHODS

The methodology followed a quantitative data analysis approach using exploratory statistical analysis methods. Comparisons were made between DOGS101and CATS101 students' patterns of usage of supporting resources using existing historical data including the number of times students used the FYPLC, the discussion forum and the automated assessment feedback submission system. Differences in gender, FYPLC and grade were examined.

ACTUAL OR ANTICIPATED OUTCOMES

There are distinct patterns of use of the supporting resources based on cohort, gender and grade. For example, DOGS101 students make more use than CATS101 students of the FYPLC. Female students in DOGS101 also tend to use the FYPLC more frequently. Female students also tend to ask more questions on the discussion forum and on average receive higher grades overall than male students. Students who obtain higher grades also tend to use the FYPLC more frequently.

CONCLUSIONS/RECOMMENDATIONS/SUMMARY

This work contributes to the literature through its exploration of how non-programmers utilise the resources available to them while learning to program. Comparisons between courses revealed distinct patterns of usage. Reflections on the different levels of student engagement are made.

KEYWORDS

Learning to program, supporting resources, patterns of engagement

Introduction

Introductory programming courses have long been regarded as notoriously difficult to teach (Cheah, 2020). These difficulties have been categorised as; insufficient metacognitive skills (such as a programming strategy or problem-solving approach); negative attitude to programming (e.g. from more senior peers or prior experience); lack of student engagement; issues with choice of language (e.g. popularity in industry versus ease of learning for novices, or the syntax itself as well as faulty mental models (e.g. in relation to variable assignment); inefficient or outdated teaching methods (such as use of static slides or books to demonstrate dynamic programming concepts); and increasing class sizes and diversity of students. Bennedsen and Caspersen (2019) found the average failure rate between 63 institutions globally (e.g. from North and South America, Europe, Africa, Australasia and Asia) to be around 33% with institutions commonly reaching 50% or higher. This is compared to an average course failure rate of 18% in all New Zealand universities (Luxton-Reilly, 2016). A systematic review by Watson and Li (2014) found that the average pass rate for most programming courses was 67.7%.

However, the imperative to teach programming to non-computer science (non-CS) majors is also acknowledged (Eccheverría et al., 2017; Lee & Lee, 2019; DeSanto et al., 2022). Graduates in engineering are increasingly expected to use programming skills to obtain insight into the data emerging from sensor technology in the oil and gas industries for example (Mohammadpoor & Torabi, 202). The inherent difficulty of programming along and the potential for non-CS majors to have less intrinsic motivation poses additional challenges in terms of learning support and could lead to increased attrition (Shell et al., 2014). Lee and Lee (2019) found that after 8 weeks of college instruction, non-CS majors were only able to perform relatively simple programming tasks and could not solve a more comprehensive problem. There have also been issues with student retention of content from CS1 (introductory computer science) courses. Early research found that 62% of CS1 graduates were unable to write a loop that computed an average (Soloway, 1984).

Students' engagement with and motivation for studying programming is also problematic. Shell et al., (2014) report that students have described not enjoying programming exercises and do not always see the connection between the exercises that they complete and course content. Prior programming knowledge has also been shown to influence students' attitudes to and success when learning to program (Chou et al., 2021). Novice student programmers were found to use rote memory, copying other students and more surface approaches to learning in contrast to students with higher levels of programming experience (Chou et al., 2021). Several other authors have found that non-majors approach learning to program differently (O'Malley, 2020; Mitchell et al., 2020). These studies utilised surveys and self-report data, or in the case of the study by Mitchell et al., (2020) relatively small cohorts of students.

In an engineering context, the introduction of a compulsory programming course has had mixed outcomes. Ramirez (2022) found that engineering students thought that programming should be compulsory, that it helped them with their problem-solving skills, but that self-confidence with programming was lower for female students. DeSanto et al., (2022) found lectures that incorporated live interactive coding activities for students, gamification and autocorrection and formative feedback during laboratories were drivers of student learning for non-CS majors.

At our institution, two introductory programming courses are offered; CATS101, and DOGS101. CATS101 is targeted at computer science, information technology, or software, electrical, or mechatronics engineering majors, where programming skills are expected to be developed throughout the program of study. It also assumes students have some intrinsic motivation to pursue the course. DOGS101 is designed to target engineering students with non-CS (chemical or civil engineering for example) or undeclared majors and was originally intended to provide students with the ability to use programming skills for modelling and to analyse data (amongst other topics). We therefore assume less intrinsic motivation and, on average, less incoming technical skills as they relate to programming. Both courses are related since DOGS101 was developed from CATS101. The two courses share much of the same content, structure, and organization. This similarity is designed to enable a student who is unsure of their major or who changes their mind to move into any major at the end of first year.

Each course offers a comprehensive suite of support in response to the inherent difficultly of introductory programming. In addition to weekly lectures, tutorials, and practical contact sessions, the following additional support options are provided for students:

1) An online discussion forum (EdStem);
2) An assessment submission platform with automated feedback (Gradescope);
3) And recently, a First Year Programming Learning Center (FYPLC).

For the reasons outlined above, providing additional resources when learning to program is important to minimise failure rates, lower attrition, instil confidence in computational thinking and problem solving and to support student learning overall. The provision of these resources can, however, lead to introductory programming courses becoming resource intensive in relation to other courses offered. This is primarily due to the volume of questions asked by students of teaching staff. To more efficiently support student learning and decrease the workload for our two introductory programming courses, we established the First Year Programming Learning Center (FYPLC). The FYPLC is a space shared between programming courses where teaching staff are available during the business hours of 9am to 5pm. It is similar to other initiatives in STEM designed to support student learning such as Mathematics Learning Centres (Dzator & Dzator, 2021; Fuller, 2002).

The ways in which non programming or computer science majors use resources designed to support their learning is of particular interest in this context. Since the two courses CATS101 and DOGS101, cater to very different cohorts, we were interested to see if there were different needs in terms of access to supporting learning resources. This research was designed to make use of existing data on student use of the FYPLC, the EdStem Discussion forum and the automated feedback system.

Research Goals and Objectives

Establishment of the FYPLC required a significant investment from both introductory programming courses. We therefore need to evaluate that the students are reaping the benefits of this new resource and compare it with benefits offered by existing resources. More generally, we seek to improve the support offered to our cohorts. Our research goals and aims were to:

- Identify resource usage patterns among the cohort.
- Identify any groups within the cohort requiring additional/alternative support.
- Find ways to improve the support offered to the cohorts

We therefore had the following research questions.

RQ1 How do the patterns of use for supporting resources differ between the two cohorts?

RQ2 What differences are there in patterns of the use of supporting resources for groups within each cohort?

Methodology

This study utilises techniques drawn from quantitative data analysis such descriptive and exploratory analyses. Ethics exemption/clearance 2023/HE001206 was granted to analyse an existing large pool of quantitative data from a variety of sources. The data sources include:

1. **Demographic data –** Gender, Course studied, Cumulative semester GPA, and Grade.
2. **Assignment submission data –** Number of times each assessment was submitted.
3. **EdStem data –** this includes, the number of questions asked, the number of questions viewed and number of days active in the EdStem Discussion forum.
4. **FYPLC usage data –** includes whether students used the FYPLC and how many times.
5. **Course data –** includes course studied (e.g. CATS101 or DOGS101) individual assessment marks as well as overall course mark.

The data we analysed was gathered during the introduction of the FYPLC in CATS101 and DOGS101 in Semester 2, 2022. During the establishing semester, CATS101 had a total of 378 students, and DOGS101 had a total of 457 students. We utilize demographic and grade data of each student. We also gather data from the online discussion forum (EdStem), assessment submission platform (Gradescope), and their participation in the FYPLC via data collected in the question queuing system.

Results

We first wish to understand the usage of the FYPLC among the respective cohorts, ensuring that it is being used, and then which demographics of students utilize the centre. Usage of the FYPLC is summarized by Table 1. In CATS101, only 31.48% of students used the FYPLC throughout the semester, while in DOGS101, the majority, 70.67% used the centre.

Comparison of usage patterns across cohorts

Frequency of use of the FYPLC by the two cohorts, CATS101 and DOGS101 are presented in Table 1. Since the data did not meet the assumptions for parametric statistical tests, the non-parametric Kruskal-Wallis test was used. Bonferroni adjustment to reported *p* values were used to control for family-wise error rates due to multiple comparisons. The data indicates that students in DOGS101 made a significantly higher number of visits to the FYPLC than did students in CATS101, $\chi^2(1) = 128.69$, $p<.001$. Female students also made a significantly higher number of visits to the FYPLC than did male students $\chi^2(1) = 23.44$, $p<.001$. Additionally, this pattern of results was found with the two cohorts of students. For example, female students within both DOGS101 and CATS101 both used the FYPLC more frequently than male students, $\chi^2(1) = 13.93$, $p<.001$ and $\chi^2(1) = 16.79$, $p<.001$, respectively.

Table 1. Frequency of use of FYPLC by the two cohorts CATS101 and DOGS101

Course code	Gender	Did not use FYPLC	Used FYPLC	Totals	Average number of uses of FYPLC
CATS101	Male	215	44	259	9.37
	Female	64	55	119	8.62
	Total	***279***	***99***	***378***	***9.13***
DOGS101	Male	103	209	312	11.01
	Female	31	114	145	15.30
	Total	***134***	***323***	***457***	***12.29***

A time series plot of the frequency of use of the FYPLC revealed that students tended to make most use of the FYPLC around assessment due dates (See Figure 1, below). For CATS101, assignments were due on the 24th of August, 21st of September and the 26th of October. For DOGS 101, the assignments were due on the 25th of August, 23rd of September and 28th of October.

It is possible that students in CATS101 used alternative supporting resources to the FYPLC. It is also possible that different subsets of students within both the CATS101 and DOGS101 cohorts utilised alternative forms of support. This finding therefore led us to explore other patterns of use by students' for the automated feedback system and EdStem discussion forum. Subsequent analysis explores this issue in more detail.

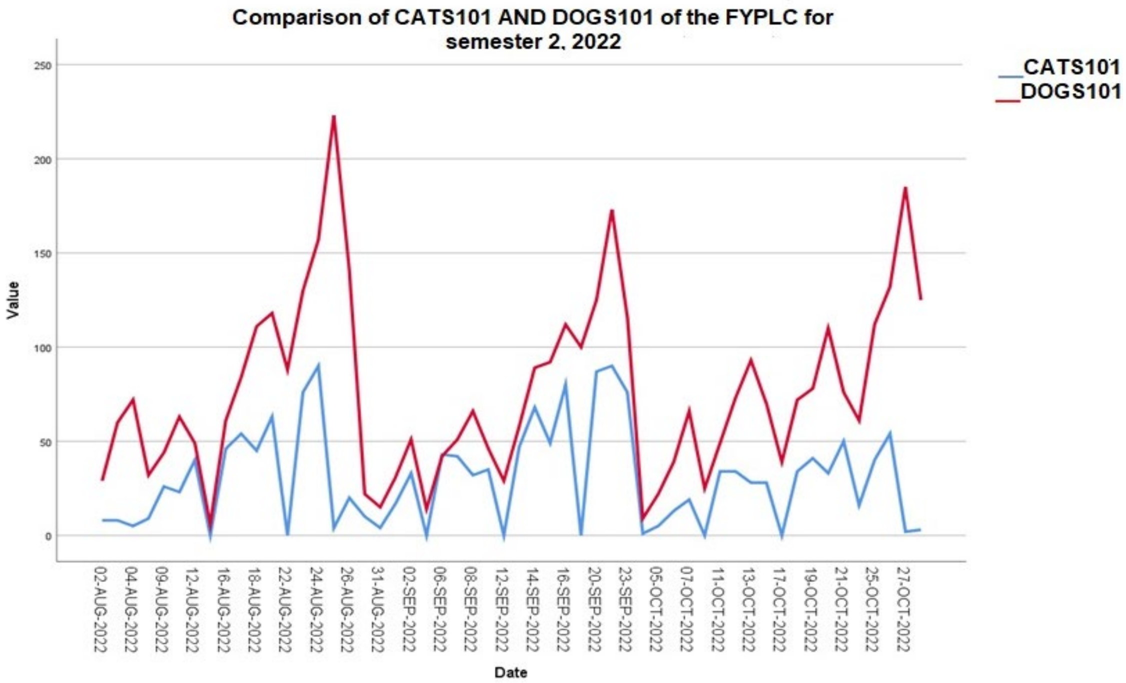

Figure 1. Time series plot for both cohorts showing peaks of use before assessment due dates.

Use of automated feedback on assignment submissions

Descriptive statistics for students' use of the EdStem Discussion Forum are shown in Table 2. The table shows that in almost every instance, students who visited the FYPLC used the Automatic assignment feedback system more than students who did not visit the FYPLC and that female students used it more than male students. The only exception is for female students in DOGS101 who visited the FYPLC; they tended to use the automated assignment feedback system less than males in the same course who had also visited the FYPLC (e.g. for Assignments 1 and submissions were $\bar{x}$= 56.81 compared to 52.85 and for Assignment 2 submissions were 59.86 compared to 57.95 respectively).

Table 2 Frequency of use of the Automated assignment feedback system by course, gender, and use of the FYPLC.

Variables	Gender	Used FYPLC	Mean	Std. Error	Mean	Std. Error
Assignment 1 Submissions	Male	No	37.57	3.51	41.80	5.07
		Yes	46.12	6.03	56.81	2.95
	Female	No	40.11	6.22	47.20	10.55
		Yes	50.27	6.42	52.85	4.08
Assignment 2 Submissions	Male	No	75.63	7.89	53.63	5.13
		Yes	130.22	13.52	59.86	2.99
	Female	No	86.28	13.95	62.67	10.68
		Yes	146.21	14.41	57.95	4.14
Assignment 3 Submissions	Male	No	31.82	2.71	44.65	5.19
		Yes	44.90	4.65	61.02	3.02
	Female	No	34.60	4.80	47.87	10.80
		Yes	57.14	4.96	62.82	4.18

Use of EdStem Discussion forum

Patterns of usage of EdStem Discussion forum data showed that students from CATS101 used the forum more than students from DOGS101 (see Table 3 for additional information). They tended to ask more questions overall, viewed the forum more frequently, and had more active days on the forum. Questions were highly varied but could include something like “Why do I get the following error message ‘IndexError: tuple index out of range’”. Female students also tended to be more active on the EdStem discussion forum than male students regardless of the cohort.

Table 3. Frequency of EdStem interaction by course, gender, and use of the FYPLC.

Variables	Gender	Used FYPLC	Mean	Std. Error	Mean	Std. Error
EdStem Questions	Male	No	1.71	0.58	0.65	1.30
		Yes	5.44	0.99	2.02	0.76
	Female	No	2.32	1.02	1.80	2.71
		Yes	5.05	1.06	3.03	1.05
Number of EdStem Views	Male	No	332.85	44.74	164.43	45.61
		Yes	560.62	76.71	252.23	26.54
	Female	No	491.00	79.12	280.47	94.94
		Yes	602.80	81.78	348.42	36.77
EdStem Days Active	Male	No	22.67	2.01	8.54	2.27
		Yes	38.82	3.44	16.21	1.32
	Female	No	25.87	3.55	15.47	4.72
		Yes	42.84	3.67	22.24	1.83

Students’ marks and cumulative GPA in CATS101 and DOGS101

Table 4 shows students’ mean course marks out of 100 and cumulative grade point average (GPA) out of 7. In general students using the FYPLC have higher average marks and GPAs than students who do not use the FYPLC. Additionally female students tend to have higher total course marks and GPAs than male students. The exception to this may be for students in CATS101 who do not use the FYPLC. In this case average grades for male and female students are almost identical for both total course marks and cumulative GPA ($\bar{x}$ = 58.32 compared 58.71 for total course mark and $\bar{x}$ = 4.19 compared to 4.25 for cumulative GPA for male and female students respectively).

Table 4 Total course mark and cumulative grade point average (GPA) by course, gender and use of FYPLC

Variable	Gender	Used FYPLC	Mean	Std. Error	Mean	Std. Error
Total Course Mark	Male	No	58.32	1.82	53.91	2.08
		Yes	68.07	3.06	69.76	1.29
	Female	No	58.71	3.22	64.89	4.42
		Yes	72.06	3.38	72.07	1.79
Cumulative GPA	Male	No	4.19	0.10	4.00	0.14
		Yes	4.85	0.17	4.93	0.09
	Female	No	4.25	0.18	4.69	0.30
		Yes	5.41	0.19	5.18	0.12

Students' patterns of use for FYPLC, EdStem and the Automated Assignment tool in relation to their total marks for each course

Two multiple regression analyses were conducted in order to develop models for each cohort (CATS101 and DOGS101). Multiple regression analysis was used to determine if FYPLC usage, EdStem usage, and use of the Automated Assessment tool significantly predicted students' final marks. The results of the regression for the two courses show:

- For CATS101, the predictors are significantly associated with students' final marks in the course. The model accounted for 53.2% of the variance (R =.730, F(1, 279) =16.11, $p < .001$).
- For DOGS101, the predictors are significantly associated with students' final marks in the course. The model accounted for 71.5% of the variance (R =.846, F(1, 363) =21.35, $p < .001$).

The process for constructing both models was the same. Variables were entered in four blocks. The first block contained the number of times the FYPLC was used, the second contained the number of submissions to the automated feedback system for assignments 1, 2 and 3, the third block contained the variables related to interaction with the EdStem discussion forum and the fourth block contained students' cumulative semester GPA. Use of FYPLC was not a significant predictor in the model for CATS101 students at the adjusted *p* value of .0083, however the addition of additional predictors in subsequent blocks of variables resulted in both models accounting for significant variance, which allows us to conclude that the independent variables were significant predictors of the dependent variable. Ozili (2023) argues that an R-square value between 0.5 and 0.99 is acceptable; in this case the R-square values are 0.532 and 0.715 for CATS101 and DOGS101 respectively.

Discussion

Students in the DOGS101, in comparison to CATS101 have significantly different patterns of use of the available support resources. They are more frequent users of the FYPLC, the automated assignment submission feedback system and the EdStem discussion forum. Additionally, female students are on average more likely to make use of the FYPLC. They also tend to use all of the other available support resources more frequently than male students regardless of which cohort they are in (with the exception of female students in DOGS101 who used the automated feedback system slightly less than male students). Female students also tended to have higher course marks and cumulative GPAs. This appears to be in part due to their increased use of the FYPLC and other supporting resources. This is indicated by very positive student comments from end of semester evaluations; an indicative quote from a student in DOGS101 being "The help centre -> I would not have passed without it and everyone there deserves a pay rise, seriously they were a godsend - (never stop offering the help centre)".

It is possible that female students more frequent use of the FYPLC and higher levels of interaction with the EdStem discussion forum was sufficient for them to address their learning needs. Alternatively, the average number of submissions to the automated assignment system could be a ceiling effect, reaching the limits of its usefulness in terms of providing guidance prior to assignment submission. However, students in CATS101 were much more frequent users of the automated assignment feedback system than students in DOGS101. This finding is surprising in some ways, until we consider that they were significantly less likely to use the FYPLC; it is possible that for students in CATS101, the automated assignment submission system provided them with learning support that was otherwise obtained by students in DOGS101 through the FYPLC.

The finding that the two cohorts of students DOGS101 and CATS101 differ in their pattern of use of supporting resources is consistent with related work which identified that non-CS majors more frequently utilize available course resources (Caleb O'Malley, 2020). It is possible that the design of the assessment in DOGS101 during the studied semester may also have impacted this increased utilization.

The implications of these findings are that students in the two cohorts (CATS101 and DOGS101) use the available support resources differently and that therefore these supports may need to be resourced differently. For example, more tutors could be placed in the FYPLC with knowledge of the DOGS101 course. Additional training could be provided to tutors on how to best support students with lower levels of prior programming experience, as well as lower likelihood of intention to pursue computer science, information technology, software, electrical, or mechatronics engineering majors. There seems to be an underlying latent or hidden variable related to *student engagement* that could partially explain these findings. Shell and Soh (2013) found lack of engagement led to ineffective learning strategies in programming, particularly for non-CS majors. This could be resolved by designing courses that develop appropriate computational thinking skills for students as well as assessments that are of the right degree of challenge (Milesi et al., 2017).

Some of the limitations of this research include a lack of data about students prior programming experience in both cohorts. This could have influenced outcomes. Additionally, it would be good to replicate these findings in other cohorts of students taking the same courses in different semesters and years. Future research could also collect data on student engagement.

Conclusion and Recommendations

The FYPLC has proved successful on a number of key measures. THE FYPLC may be addressing some of novice programmers limitations of cognitive load in working memory, novices tend to learn skills very effectively in the first instance by studying examples. In the FYPLC students can learn from the example of tutors as they solve problems, evaluate coding alternatives and even construct simple code fragments. The success of the FYPLC has been such that traditional “pracs” have been discarded in DOGS101 for the past two semesters and there has been no student dissent for this change.

This study shows how existing quantitative data can be utilised to explore the patterns of use of resources to support student learning for both programming and non-programming majors. This data shows that certain students need to engage with this type of learning sooner rather than later. Those completely new to programming and males in particular, need encouragement in this regard. To this end, it is recommended that at the beginning of the semester, a past student be co-opted to relate their past experiences to the commencing cohort. This has already been trialled once in DOGS101, where a male student recounted how much better he progressed once he started availing himself of the tutor assistance. Understanding the role of engagement in learning to program could help us to develop supporting resources tailored to assist those students that do not appear to make effective use of any of the supporting resources provided.

The role of good course design in supporting non-programming students to learn to program is also highlighted by this research. This includes reinforcing to students the utility and application of programming concepts in their other disciplinary studies as well as to their future careers.

References

Bennedsen, J. and Caspersen, M. E. (2019). *Failure rates in introductory programming: 12 years later.* ACM Inroads, 10(2):30-36.

Caleb O'Malley. (2020). *How do Non-Majors Approach a CS1 Course?* In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), ep272.

Chou, T. L., Tang, K. Y., & Tsai, C. C. (2021). A Phenomenographic Analysis of College Students' Conceptions of and Approaches to Programming Learning: Insights From a Comparison of Computer Science and Non-Computer Science Contexts. *Journal of Educational Computing Research*, *59*(7), 1370-1400.

De Santo, A., Farah, J. C., Martínez, M. L., Moro, A., Bergram, K., Purohit, A. K., ... & Holzer, A. (2022). Promoting Computational Thinking Skills in Non-Computer-Science Students: Gamifying Computational Notebooks to Increase Student Engagement. *IEEE Transactions on Learning Technologies*, *15*(3), 392-405.

Dzator, M., & Dzator, J. (2020). The impact of mathematics and statistics support at the Academic Learning Centre, Central Queensland University. *Teaching Mathematics and its Applications: An International Journal of the IMA*, *39*(1), 13-28.

Echeverría, L., Cobos, R., Machuca, L., & Claros, I. (2017). Using collaborative learning scenarios to teach programming to non-CS majors. *Computer applications in engineering education*, *25*(5), 719-731.

Fuller, M. (2002). The role of mathematics learning centres in engineering education. *European journal of Engineering education*, *27*(3), 241-247.

Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 7).

Lee, G., & Lee, E. H. (2019, November). Computational Thinking and Programming for Non-CS Majors. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 1013-1018). Association for the Advancement of Computing in Education (AACE).

Luxton-Reilly, A. (2016). *Learning to program is easy.* In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE'16, page 284-289, New York, NY, USA. Association for Computing Machinery.

Mohammadpoor, M., & Torabi, F. (2020). Big Data analytics in oil and gas industry: An emerging trend. *Petroleum*, *6*(4), 321-328.

Mehmood, E., Abid, A., Farooq, M. S., & Nawaz, N. A. (2020). Curriculum, teaching and learning, and assessments for introductory programming course. *IEEE Access*, *8*, 125961-125981.

Milesi, C., Perez-Felkner, L., Brown, K., & Schneider, B. (2017). Engagement, persistence, and gender in computer science: Results of a smartphone ESM study. *Frontiers in psychology*, *8*, 602.

Mitchell, S., Cole, K., & Joshi, A. (2020, March). X+ CS: A Computing Pathway for Non-Computer Science Majors. In *ASEE Mid Atlantic Section Spring Conference, 2020*.

Muresano, R., Rexachs, D., & Luque, E. (2010). Learning parallel programming: a challenge for university students. *Procedia Computer Science*, *1*(1), 875-883.

Ozili, P. K. (2023). The acceptable R-square in empirical modelling for social science research. In *Social research methodology and publishing results: A guide to non-native english speakers* (pp. 134-143). IGI Global.

Ramirez, J. (2022). The Impacts of a New Programming Course in a First-Year Engineering Experience.

Shell, D. F., & Soh, L. K. (2013). Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses. *Journal of Science Education and Technology*, *22*, 899-913.

Soloway, E. (1984). *What do novices know about programming.* Directions in Human-Computer Interaction.

Copyright statement