Full Paper - Building on the First-Year Engineering programming experience: Understanding the motivation and self-efficacy of students in a follow-on programming course

Dr. Joseph A Lyon, University of Notre Dame

Joseph A. Lyon is an Assistant Teaching Professor at the University of Notre Dame. He holds a Ph.D. in Engineering Education. His research interests are computational thinking and mathematical modeling.

Dr. Mayari I. Serrano, Purdue University at West Lafayette (PPI)

Mayari Serrano Anazco is a visiting clinical assistant professor in the Honors College at Purdue University. She earned her Bachelor's degree in Biotechnology Engineering at Ecuador's Army Polytechnic School and her Master's and Ph.D. degrees in Computer and Information Technology from Purdue University. In 2018, she and Dr. Suzanne Zurn-Birkhimer and Dr. Beth M. Holloway were conferred the Susan Bulkeley Butler Research Fellowship Award. After obtaining her Ph.D., she was appointed as the first post-doctoral fellow of the Women in Engineering Program at Purdue University. Mayari Serrano has worked towards increasing women's participation in technology and engineering for over eight years previous coming to the John Martinson Honors College. She has authored, co-authored, implemented, and assessed learning activities, outreach activities, and workshops focused on modifying negative attitudes towards technology and engineering and increasing knowledge of several topics of STEM (science, technology, engineering, and mathematics). Mayari Serrano's research focuses on commercial and educational technologies' effect on teaching, learning, diversity, and inclusion. Her publications show interdisciplinary interest and cover multimodal learning environments, embodied cognition, complex concepts, user experience, spatial abilities, gender bias, gender stereotypes, e-mentoring, sense of belonging, campus climate, and exergaming.

Full Paper - Building on the First-Year Engineering programming experience: Understanding the motivation and self-efficacy of students in a follow-on programming course

Introduction

Computing continues to be of paramount interest to engineering researchers and educators who are looking to produce computational-thinking-enabled professionals for the workforce [[1], [2]. And much of this development begins as early as first-year engineering for many students, where programming is a common component to introductory engineering courses [3], [4]. However, the development of computational thinking and programming needs to continue beyond the first year and spiraled throughout the curriculum [5], [6].

First-year engineering computation can come in many different forms. For example, many engineering programs both in the United States as well as globally teach a programming language in their first-year sequence of classes, often focused on MATLAB or Python [3], [7], [8], [9]. Sometimes this is also paired with physical computing devices such as microcontrollers or processors such as Raspberry Pi or Arduino [10], [11]. However, these skills continue to be undertaught after the first year for a variety of reasons including limited curriculum space as well as faculty resistance.

This has led to a host of different factors, including students undervaluing the importance of programming as well as struggling to feel confident or able in their computing abilities [4], [12]. This had led the research team to two exploratory questions regarding a follow-on programming course: (1) In what ways do students value a follow-on programming course to their first-year engineering experience? (2) How are students' motivation and self-efficacy related after a follow-on programming course to their first-year engineering experience?

Background

Programming in First-Year Engineering Education
Introductory programming is very prevalent among engineering programs across the United States and globally [8], [12], [13], [14]. Oftentimes students find the programming content to be extremely frustrating [15] leaving the students with less than optimistic views of programming or believing it to not be as valuable as other aspects of their introductory engineering course [12]. This can often lead to mixed results as to whether students feel comfortable programming after an introductory experience with the material [14]. This study aims to use a follow-on course to help lower the frustration, create a more open environment to learn, and situate the learning in relevant engineering context so that students see the usefulness of their first-year introductory programming experiences.

Course Design

The course design was simple given it was a one-credit hour class, meeting for one hour a week over one semester. It was in a seminar style, where one new method for using computer programming in the context of engineering was learned each week. An overview of the semester and topics covered is shown in Table 1.

Table 1. Overview of the semester topics and structure

Week	Topic	Assignment
1	Introduction to Numerical Methods	Pick numerical method to present to class
2	Python/MATLAB Refresher	Practice problems
3	Newton-Raphson Root Finding	Student Presentation, Practice Problem
4	Numerical Differentiation	Student Presentation, Practice Problem
5	Numerical Integration	Student Presentation, Practice Problem
6	Linear Regression	Student Presentation, Practice Problem
7	Non-linear Regression	Student Presentation, Practice Problem
8	Linear Optimization	Student Presentation, Practice Problem
9	Monte Carlo Methods	Student Presentation, Practice Problem
10	Euler’s Method	Student Presentation, Practice Problem
11	Finite Difference Method	Student Presentation, Practice Problem
12	Finite Element Analysis	Student Presentation, Worktime for final report
13	No additional presentation	Worktime for final report
Thanksgiving Break – No Class		
15	Summary/Course Wrap Up	Turn in final report
Quiet Week – No Class		

Each topic was given a week, where a student or team of students worked together to present that topic to the class for 30 minutes, including how to use the method and what kinds of uses it would have in engineering or industry more broadly. At the end of each week, there would be an example problem for the students to solve. These would often be easy but aimed at getting the students to better understand the usefulness. See one example in Figure 1 below.

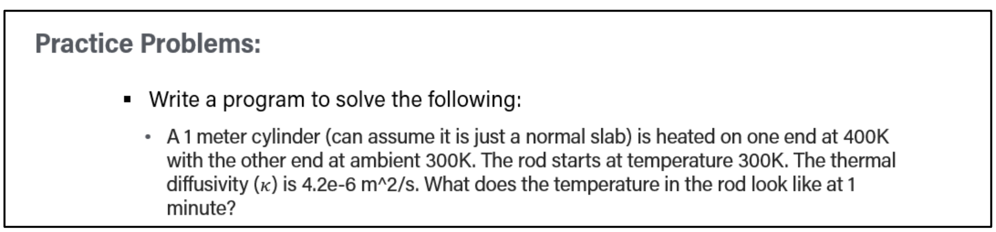

Figure 1. Example of simple computational problem at the end of a one-hour lecture

In addition to the practice problems and the presentation in class, each student had to write a report that detailed either: (1) two different methods from the class used to solve real-world problems from their anticipated career trajectory or (2) a detailed report on one of the methods and how it was used normally in their anticipated career trajectory (i.e. how did they foresee themselves using these methods and what problems in their field would they be working on).

Methods

Participants

The study's participants were students enrolled in an elective seminar course aimed at continuing to learn engineering programming. The students were all in the fall semester following their first-year engineering experience. The first-year engineering experiences of these students included learning both Python and MATLAB in a team environment. No demographic data was collected as part of the study, which we acknowledge limits some of the generalizations that can be made from the study [16].

Data Collection

Data was collected as part of a follow-on programming course taught by the lead researcher through end-of-class surveys. The study procedures for data collection were approved via the Institutional Review Board and participation in the study was completely voluntary. All students on the course participated, allowing for a sample of nine students for the study. The survey was given near the end of the class when students completed most of the follow-on programming course.

The survey instruments used for the study were obtained from previous studies in literature. First, a survey looking at the self-beliefs and self-efficacy of students in regard to computational thinking was used to understand the self-efficacy of students in this class was used from the literature [17]. Additionally, the MUSIC model of motivation survey was used to measure various motivation constructs after taking the course [18], [19]. The MUSIC model of motivation survey can be found in the literature [19]. All statements were based on a 6-points Likert scale ranging from strongly disagree (1) to strongly agree (6). One qualitative question was also asked of the students: how has being part of this class changed your perspective on computer programming and computation?

Data Analysis

The data was analyzed using two approaches. First, the quantitative data was analyzed using simple descriptive statistics and correlation coefficients to understand if the two measures of motivation and self-efficacy may be related. Additionally, a thematic analysis process was used to analyze the qualitative data responses from the students [20].

Results

How are students' motivation and self-efficacy/beliefs related after the follow-on course?

Table 2 overviews the survey's measures in an aggregate of the students' self-efficacy and beliefs and the motivation metrics. Definitions for what was considered a high correlation were obtained from the literature [21].

The results, while exploratory in nature and small in sample size, show that self-efficacy and motivation are related with the two having a strong correlation (0.97). As expected, given the overlap of some of the metrics and questions, it appears that certain metrics such as usefulness, success, and interest had the closest relationship with the self-efficacy and beliefs questions. More surprisingly, the metric of how caring they felt the instructor/environment was also moderately correlated with the self-efficacy and beliefs of the student. In general, it is encouraging to see scores for both self-efficacy and motivation were high across the entire student set.

Table 2. Average and correlation of self-efficacy/belief and motivation metrics

Student	Self-Efficacy	Empowerment	Usefulness	Success	Interest	Caring	Total Motivation
C	6.00	6.00	6.00	6.00	6.00	6.00	6.00
A	5.75	6.00	6.00	6.00	5.83	6.00	5.97
B	5.63	6.00	6.00	6.00	5.00	6.00	5.80
F	5.50	5.80	5.60	5.75	5.50	5.83	5.70
G	5.25	5.20	5.40	5.75	5.33	5.67	5.47
I	5.25	5.00	4.80	6.00	5.33	5.83	5.39
H	4.50	5.60	4.60	5.75	5.17	5.17	5.26
D	4.38	5.20	4.80	4.00	5.17	5.83	5.00
E	3.75	4.40	3.80	4.00	4.00	5.33	4.31
Correlation (r):		0.81**	0.94**	0.84**	0.84**	0.78*	0.97**

**Very strong correlation (0.8<r<1)
*Moderate correlation (0.6<r<0.8)

What do students value in the follow-on course?
Three themes of what students valued from the course were identified through the open response questions. In addition to a name and description of each theme, we have included what percentage of the students' responses that this code was within.

(1) Solving complex/unsolvable problems (5 students): The students discussed in detail that they saw the value of computer programming in solving problems that would otherwise be unsolvable. One student noted that:

> *I have learned how many different ways there are to solve complex problems that would otherwise require immense amounts of power or could be impossible, which has made me appreciate computation more.*

The students appreciated that the course connected how the programming was useful above other types of methods. For example, the course discussed how the finite element method allows for engineers to solve problems around incredibly complex geometries, situations where an analytical solution is if not entirely impossible.

It also brought an appreciation from the students as to the value of the content. One student wrote, "*It's made me realize how hard all these computations would've been before computers.*" This quote, among others, demonstrates that the students were really opened to the value of computer programming, something they may not have seen in their first-year courses.

(2) Usefulness/application to future career (2 students): One thing that the students highlighted multiple times was the overall usefulness they saw in the content to their future careers. Many students felt coming into the course that programming had limited uses in engineering. One student wrote that:

This class has shown me more uses of computer programming than just software creation. Furthermore, I've found that its applications overlap with my studies more than I had originally realized.

This opening up of what students see as the value of computer programming, as well as the importance, was evident in the data. Students noticing the prevalence throughout the fields of engineering, where one student wrote:

This class has taught me how common it is for computer analysis to be computationally and not based purely through derivation.

This idea that programming is not solely limited to software engineering, or the field of electrical and computer engineering is a benefit and value that the students picked up.

(3) Practicing computation (2 students): A few students also mentioned the value of having a space to practice computation. Given that many learn it during their first year but do not see it again for some time, students mentioned that practice was important for them and their retention, one student wrote that "[the class] helped me practice it. Otherwise, I would lose skills." Many students, if they understand the importance of the topic, want to continue to learn and work on these skills.

Discussion

While exploratory in nature, this study does add to previous work showing that computation, when put into engineering context, can be a useful way to present these abstract ideas so that students better learn underlying computing principles [22], [23], [24]. However, these results are even more encouraging because students seemed to really notice the value of computation when presented in this way, and students understanding value is linked to achievement emotions and, motivation [25].

Our quantitative survey results did show some promising links between self-efficacy and beliefs along with motivation as measured by the MUSIC model of motivation [18], [19]. The fact that these two factors are related has been extensively shown in the literature previously [26], [27], [28]. Our results simply continue to show that these two factors, within the context of a follow-on programming course, continue to hold. Additionally, our findings use the MUSIC model of motivation to break out the different smaller pieces of motivation: empowerment, usefulness, success, interest, and caring and look at the relationship with self-efficacy and belief, showing that there were significant relationships between each subcategory of the MUSIC model with self-efficacy and belief.

Finally, our results, while only a small sample size, show that this type of low-stress seminar approach to a programming class as a follow-on to first-year engineering programming could prove extremely useful in opening students to programming as it relates to their future engineering discipline. Overall student motivation and self-efficacy was high across nearly all the questions and students.

Conclusion and Limitations

To conclude, using a follow-on programming course where students learn programming methods in the context of engineering problems seemed to be successful in, at least, maintaining student motivation as well as self-efficacy and beliefs. However, our study has several limitations that need to be further analyzed. First, a small sample size as well as lack of demographic information on the students limits how generalizable the results are. Second, the lack of a pre-test measurement limits what this study has to say about how the students changed throughout the semester. The authors hope to continue to investigate how to best use either a follow-on or supplemental course for programming during the first-year engineering experience.

References

[1] A. J. Magana and G. Silva Coutinho, "Modeling and simulation practices for a computational thinking-enabled engineering workforce," *Computer Applications in Engineering Education*, vol. 25, no. 1, pp. 62–78, 2017, doi: 10.1002/cae.21779.

[2] J. Malyn-Smith and I. Lee, "Application of the Occupational Analysis of Computational Thinking-Enabled STEM Professionals as a Program Assessment Tool," *Journal of Computational Science Education*, vol. 3, no. 1, pp. 2–10, 2012, doi: 10.22369/issn.2153-4136/3/1/1.

[3] L. M. Cruz Castro, A. J. Magana, K. A. Douglas, and M. Boutin, "Analyzing Students' Computational Thinking Practices in a First-Year Engineering Course," *IEEE Access*, vol. 9, pp. 33041–33050, 2021, doi: 10.1109/ACCESS.2021.3061277.

[4] G. M. Hutchison, M. A., Follman, Deborah K, Sumpter, Melissa, Bodner, "Factors Influencing the Self-Efficacy Beliefs of First-Year Engineering Students.," *Journal of Engineering Education*, vol. 95, no. 1, pp. 39–47, 2006.

[5] J. A. Lyon and A. J. Magana, "Computational thinking in higher education : A review of the literature," *Computer Applications in Engineering Education*, pp. 1–16, 2020, doi: 10.1002/cae.22295.

[6] J. A. Lyon and A. J. Magana, "A Review of Mathematical Modeling in Engineering Education," *International Journal of Engineering Education*, vol. 36, no. 1, pp. 101–116, 2020.

[7] S. Canfield and M. Abdelrahman, "Enhancing the Programming Experience for First-Year Engineering Students through Hands-On Integrated Computer Experiences," *Journal of STEM Education*, vol. 13, 2012.

[8] S. Nikolic, M. Ros, and D. B. Hastie, "Teaching programming in common first year engineering: discipline insights applying a flipped learning problem-solving approach," *Australasian Journal of Engineering Education*, vol. 23, no. 1, pp. 3–14, Jan. 2018, doi: 10.1080/22054952.2018.1507243.

[9] J. A. Lyon and T. M. Whalen, "WIP: Designing disciplinary projects in an honors first-year engineering course to improve retention and participation of first-year students. WIP: Designing multidisciplinary projects in an honors first-year course to broaden students' conception of engineering," in *Proceedings of the 2023 ASEE Conference*, 2023.

[10] W. Lawson, S. Secules, and S. Bhattacharyya, "An application-based freshman introductory programming course using the Raspberry Pi," *First Year Engineering Experience Conference*. Wiley-Blackwell Publishing Ltd, 2016. doi: 10.1002/j.2168-9830.2004.tb00809.x.

[11] H. Smith, "Microcontroller Based Introduction to Computer Engineering," in *7th First Year Engineering Experience Conferece*, 2014. [Online]. Available: www.atmel.com

[12] H. Murzi, V. Tech, J. Jurkiewicz, and K. Reid, "First-year engineering program evaluation: Understanding senior students' perceptions about their first-year experience," in *Proceedings of the 2020 ASEE Conference*, 2020.

[13] S. R. Sobral, "Teaching and learning to program: Umbrella review of introductory programming in higher education," *Mathematics*, vol. 9, no. 15, Aug. 2021, doi: 10.3390/math9151737.

[14] J. E. Lewis, B. S. Robinson, and N. Hawkins, "First-Year Engineering Student Perceptions in Programming Self-Efficacy and the Effectiveness of Associated Pedagogy Delivered via an Introductory, Two-Course Sequence in Engineering," in *The Proceedings of the 2020 ASEE Conference*, 2020.

[15] Z. Atiq and M. C. Loui, "A Qualitative Study of Emotions Experienced by First-year Engineering Students during Programming Tasks," *ACM Transactions on Computing Education*, vol. 22, no. 3, Jun. 2022, doi: 10.1145/3507696.

[16] A. L. Pawley, "Shifting the 'Default': The Case for Making Diversity the Expected Condition for Engineering Education and Making Whiteness and Maleness Visible," *Journal of Engineering Education*, vol. 106, no. 4, pp. 531–533, 2017, doi: 10.1002/jee.20181.

[17] A. J. Magana, M. L. Falk, C. Vieira, and M. J. Reese, "A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices," *Comput Human Behav*, vol. 61, pp. 427–442, 2016, doi: 10.1016/j.chb.2016.03.025.

[18] B. D. Jones and J. L. M. Wilkins, "Testing the MUSIC Model of Academic Motivation through confirmatory factor analysis," *Educ Psychol (Lond)*, vol. 33, no. 4, pp. 482–503, Jul. 2013, doi: 10.1080/01443410.2013.785044.

[19] B. D. Jones, "Motivating Students to Engage in Learning: The MUSIC Model of Academic Motivation," *International Journal of Teaching and Learning in Higher Education*, vol. 21, no. 2, pp. 272–285, 2009, [Online]. Available: http://www.isetl.org/ijtlhe/

[20] V. Braun, V. Clarke, V. Braun, and V. Clarke, "Using thematic analysis in psychology Using thematic analysis in psychology," *Qual Res Psychol*, vol. 0887, no. 2006, 2006.

[21] Y. H. Chan, "Biostatistics 104: Correlational Analysis," 2003.

[22] A. J. Magana, M. L. Falk, and M. J. Reese Jr., "Introducing Discipline-Based Computing in Undergraduate Engineering Education," *ACM Transactions on Computing Education*, vol. 13, no. 4, pp. 16:1-16:22, 2013, doi: 10.1145/2534971.

[23] J. A. Lyon and A. J. Magana, "The use of engineering model-building activities to elicit computational thinking : A design-based research study," *Journal of Engineering Education*, pp. 1–23, 2021, doi: 10.1002/jee.20372.

[24] J. A. Lyon, A. J. Magana, and M. Okos, "WIP: Designing modeling-based learning experiences within a capstone engineering course," *ASEE Annual Conference and Exposition*, 2019.

[25] R. Pekrun, A. C. Frenzel, T. Goetz, and R. P. Perry, "The control-value theory of achievement emotions. An integrative approach to emotions in education.," in *Educational psychology series. Emotion in education.*, San Diego, CA: Elsevier, 2007, pp. 13–36.

[26] D. H. Schunk, "Self-efficacy, motivation, and performance," *J Appl Sport Psychol*, vol. 7, no. 2, pp. 112–137, 1995, doi: 10.1080/10413209508406961.

[27] D. H. Schunk, "Social cognitive theory and motivation," in *The Oxford Handbook of Human Motivation* , 2012, pp. 11–26.

[28] M. Prat-Sala and P. Redford, "The interplay between motivation, self-efficacy, and approaches to studying," *British Journal of Educational Psychology*, vol. 80, no. 2, pp. 283–305, Jun. 2010, doi: 10.1348/000709909X480563.