Examining Self-Efficacy and Growth Mindset in an Introductory Computing Course

James J Pembridge, Ph.D.
Engineering Fundamentals
Embry-Riddle Aeronautical University
Daytona Beach, FL, USA
pembridj @erau.edu

Kelsey J Rodgers, Ph.D.
Engineering Fundamentals
Embry-Riddle Aeronautical University
Daytona Beach, FL, USA
rodgerk6@erau.edu

***Abstract*—This work-in-progress research paper examines student perceptions of self-efficacy and growth mindset associated with computer programming in a first-year engineering computing course. While prior research has begun to explore the general relationship between student self-efficacy and mindset, it has demonstrated that context matters greatly for these constructs. This study builds on that research by examining both constructs within the context of an introductory programming course for engineers and addressing the following research questions: 1) How do student perceptions of programming self-efficacy relate to programming mindset? 2) What types of previous experiences in programming are associated with a growth mindset? The mixed methods study presented in this paper examines self-reported perceptions from students enrolled in a first year introduction to engineering computing course that utilized MATLAB as the programming language. Findings indicate that within this setting, students were generally confident in their ability to program in MATLAB and had a slight growth mindset towards programming. However, there was no significant correlation between the two constructs. Additionally, students who were retaking the course had a higher growth mindset towards programming as a talent. No other mastery experiences or vicarious experiences were associated with overall growth mindset.**

Keywords—self-efficacy, growth mindset, computer programming, first-year engineering

I. Introduction

The first year of engineering students' undergraduate education presents many opportunities to significantly impact students' remainder of their education and retention in engineering programs. Encouraging growth mindset and ensuring positive impacts on students' self-efficacy are two ways to positively impact engineering students' experiences [1-3]. Therefore, this study seeks to examine the following research questions:

RQ1. How do student perceptions of programming self-efficacy relate to programming mindset?

RQ2. What types of previous experiences in programming are associated with growth mindset?

II. Literature Review

This study looks at two beliefs about one's own abilities – self-efficacy and growth mindset.

A. Self-Efficacy

Self-efficacy is the belief in one's own ability to successfully complete a task or accomplish something. Self-efficacy theory states that if someone believes they will succeed in something, they will attempt it and in reverse if someone believes they will fail in something, they will avoid it [4]. Self-efficacy is a self-judgement of one's own abilities; it is not a personality trait that can be measured in a general since, but rather a belief dependent on the task or domain [5]. Elementary programming courses offer a critical opportunity to assess self-efficacy as these courses are well known for low retention rates and perceived difficulty [6].

Bandura [5] proposed four principal sources that a person derives personal efficacy from, which he later further built upon. Bandura discussed the four principal sources of self-efficacy beliefs of: (1) enactive mastery experience, (2) vicarious experience, (3) verbal persuasion, and (4) physiological and affective states [7]. Enactive mastery experience is the most influential source of efficacy because it requires authentic experiences where someone succeeds or fails. Success improves efficacy, while failure diminishes it; although overcoming obstacles also builds efficacy. Vicarious experience refers to social comparisons and observations. Verbal persuasion is about the feedback, acknowledgment, and appraisal someone receives about their abilities. Physiological and affective states refer to physical responses (e.g., sweating, windedness) and mood states (e.g., stressed, excited) [7].

B. Growth Mindset

Mindset is a recognized set of attitudes and beliefs about one's own personal traits that can be changed. Growth mindset is a belief that one's own ability and intelligence can be fostered through hard work and commitment [8]. A fixed mindset is an opposing mindset; it is a belief that one's own talent and intelligence are fixed traits, where one must discover their abilities rather than develop them [8].

Having a growth mindset positively correlates to academic performance and teaching a growth mindset has been found to positively impact students' academic performances [e.g., 1, 8, 9]. Messages around teaching growth mindset focus on intelligence being malleable and developed through practice. Most education applications of teaching growth mindset focus on intelligence.

Dweck theorized that people can have different mindsets about intelligence, talent, and/or athletic ability [8, 10]. Although there are no theories that justify the existence of

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

domain-specific mindsets, there are some studies that suggest that domain-specific mindsets may exist. Devers [9] discussed anecdotal evidence that students discussed fixed mindset ideas about electricity and magnetism in class, which conflicted with their growth mindset determined by their responses to the growth mindset instrument. Scott and Ghinea [2] conducted research to compare the beliefs about intelligence and programming aptitude for students in an introductory computer programming course. They found that most students with the fixed belief for programming aptitude had the growth belief for intelligence and suggest that there are domain-specific mindsets, at least for computer programming [2].

III. METHODS

The proposed research questions were studied using an exploratory mixed-methods design that relied on student self-reported perceptions of self-efficacy and growth mindset associated with computer programming while enrolled in a first-year engineering computing course.

A. *Setting and Participants*

This study was conducted within a first-year engineering department at a medium-sized, moderately research active university. Participants included students enrolled in three different sections of an introduction to computing for engineers course during the Spring semester. Overall there were 67 students enrolled across the three sections; 81% of the students were enrolled in the College of Engineering with the remaining 19% enrolled in other STEM focused colleges, 79% of the students were male, and 52% were classified as freshman, 37% sophomore, and the remaining were undergraduate upper level students.

When examining the self-reported survey items, 42% of the respondents indicated they had taken the course previously. Out of the remaining 58% who were taking the course for the first time, 42% of them had previous experience with programming prior to taking the course.

B. *Data Collection*

During the first week of the course, students completed reflective responses in class regarding their perceptions and experiences with programming languages. The students were prompted to:

> "Please write a reflection about any experiences you have related to programming, building computers, problem solving, etc. Please be sure to include if you have taken this class before, if you have coded in any other language before (and what language/s), any robotics experiences you may have, and any other things you would like me to know about you."

All of the students enrolled in the course completed the reflections (n = 67).

During the second week of the course, prior to any course exams or major assignments, students completed a survey that included measures of computer programming self-efficacy, a measure of growth mindset, and student demographic information. Overall, the survey received a 94% response rate (n = 63).

The self-efficacy instrument was adapted from a survey developed by Ramalingam and Wiedenbeck [6] that examined the programming self-efficacy of students programming in C++. The designed survey contained 32 items that loaded onto 4 factors that explained 65.5% of the variance among the items after rotation [6]. The four factors included independence and persistence, complex programming tasks, simple programming tasks, and self-regulation.

Since the original survey by Ramalingam and Wiedenbeck [6] was developed using language associated with C++, the language in the survey was modified to reflect the course use of MATLAB. To ensure the validity of the survey after the changes in terminology, a confirmatory factory analysis and reliability analysis were run on the 4 factors. Overall the instrument explained 78% of the variance and had an overall Cronbach reliability of .979 with individual factors having a reliability greater than .915 (Table I).

TABLE I. CRONBACH RELIABILITY OF SELF-EFFICACY FACTORS

Factor	**Cronbach Reliability**
Independence & Persistence	.955
Complex Programming Tasks	.952
Simple Programming Tasks	.915
Self-Regulation	.938

The growth mindset instrument was based on Dweck's [11] 16 item Likert survey, that Devers [9] used to examine classroom performance of physics students to their academic performance. The survey asked questions that represented both fixed and growth mindset perspectives on intelligence and talent. The Likert items consisted of 6 categories ranging from "strongly disagree" to "strongly agree". All fixed mindset items were reverse coded so that strongly agree was associated with a growth mindset.

Since other researchers argue that mindset is contextual within a specific domain, especially for computer programming [2], the survey in this study prompted students to focus their responses on "intelligence associated with programming and MATLAB only" at the beginning of the 16 questions. An exploratory factor analysis of the student responses indicated two factors (intelligence and talent) that explained 66% of the variance in the items. Overall the instrument had a Cronbach reliability of .927.

C. *Data Analysis*

Research question 1, "How do student perceptions of programming self-efficacy relate to programming mindset?" was analyzed using descriptive statistics and non-parametric correlation analyses. Non-parametric statics were used due to the relatively small number of participants and the use of Likert items in the survey instruments.

Research question 2, "What types of previous experiences in programming are associated with growth mindset?" was statistically analyzed by comparing the survey responses of those who were retaking the course and those who were taking it for the first time. Additional analyses of these question involved the qualitative analysis of the reflective responses to see what type of experiences that had.

IV. Results and Discussion

A. *Descriptive survey responses.*

Over half of the participants indicated that they were at least fairly confident in their ability to complete simple tasks, maintain self-regulation, and their interdependence and persistence. They were slightly less confident in their ability to solve complex programming tasks. The average scores for three factors were between 4.5 and 5, while the average for complex programming tasks were below 4. The standard deviations for all four factors were around 1.5 (Fig. 1).

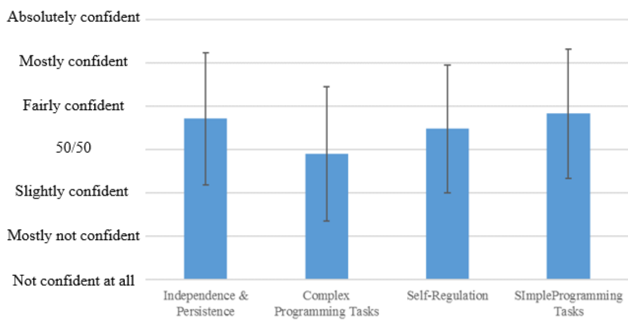

Fig. 1. Average and standard deviation of self-efficacy responses broken out by the respective instrument factors

Similar responses were observed regarding the growth mindset items, where overall less than 15% of the participants trending towards an average fixed mindset. This fixed mindset perspective is enhanced when examining the questions associated with talent (in comparison to intelligence). When addressing talent, 24% of the respondents had a fixed mindset. The average score for the total, intelligence, and talent growth mindset were slightly above mostly agree. The standard deviations ranged from 0.9 to 1.1 (Fig. 2).

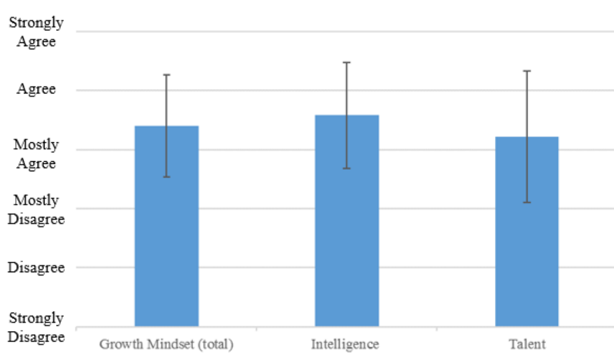

Fig. 2. Average and standard deviation of growth mindset responses by intelligence and talent factors

B. *Correlation between self-efficacy and growth mindset*

There were no statistical correlations between the growth mindset and self-efficacy instruments. Even without significance it is important to note that many of the self-efficacy constructs had either a negative correlation or no correlation to growth mindset (Table II).

TABLE II. Correlation bewtween Self-Efficacy and Growth Mindset Instruments

	Growth Mindset *(total)*	**Intelligence**	**Talent**
Independence & Persistence	-0.03	0.07	-0.07
Complex Programming Tasks	-0.23	-0.12	-0.23
Simple Programming Tasks	-0.12	-0.05	-0.11
Self-Regulation	-0.03	0.03	-0.03

C. *Comparison of prior programming experiences to growth mindset*

The first approach to comparing the impact of prior experiences involved grouping the survey responses into two groups: those who were re-taking the course (1) and those who were taking for the first time (0). A Mann-Whitney test indicated that there was no statistically significant difference between those who were re-taking the course and those who were taking the course for the first time for either the self-efficacy constructs (Table III) or the growth mindset constructs (Table IV).

TABLE III. Comparison of Self-Efficacy Median Scores between Repeating and First Time Students

	Median Score			
	First time	**Repeat**	**U**	**p**
Independence & Persistence	4.8	5.0	387	0.169
Complex Programming Tasks	3.7	4.1	446	0.578
Simple Programming Tasks	4.4	5.7	391	0.187
Self-Regulation	4.8	5.0	410.5	0.293

TABLE IV. Comparison of Growth Mindset Median Scores between Repeating and First Time Students

	Median Score			
	First time	**Repeat**	**U**	**p**
Growth Mindset (total)	4.2	4.5	396.5	0.213
Intelligence	4.5	4.9	449.5	0.612
Talent	4.0	4.1	378	0.133

The only differences between the two groups were observed for two specific items in the talent construct for the growth mindset instrument. The item level Mann-Whitney U test of "You can always substantially change how much talent you have" indicated that the growth mindset was greater for those who had taken the class before (Median = 5) in comparison to those who were taking the course for the first time (Median = 4), (U= 283.5, p =.004). The same result was seen for the item ""You can change even your basic level of talent considerably" (U = 326.5, p=.022).

To further explore how student experiences related to the growth mindset score, student reflections were qualitatively analyzed [12] for themes associated with Bandura's principle sources of self-efficacy: enactive mastery experiences, vicarious experiences, verbal persuasion, and physiological and affective states [7]. For each of the sources, student responses were categorized as either non-mentioned, negative experience, and positive experience. Every student stated their experience with programming to address the prompt, so students that stated they had no experience were categorized as non-mentioned for mastery experience.

Most of the students mentioned some sort of enactive mastery experience (65%) with some aspect of programming, which closely aligned with the survey responses. These mastery experiences were either positive which were typically prior coding experiences in other programming languages or negative experiences, which were most commonly the student failed the course in a previous semester.

Only 16% of the participants (n = 10) noted vicarious experiences. Some negative vicarious experiences were seeing their peers struggle in and/or fail this class and hearing "horror stories" of the class. Only two students had positive vicarious experiences; one student heard coding is a lot of fun once you understand it and another student watched her parent code in another programming language.

No students indicated any mention of verbal persuasion.

About a third of the students (32%) mentioned physiological responses to or affective states about programming. Half of these students mentioned a positive response or state and the other half a negative one. Positive states mentioned were enjoyment, excitement, eagerness, and invigorated. Negative states mentioned were confused, frustrated, stressed, worried, nervous, annoyed, and fearful.

Since there were no correlations between self-efficacy and growth mindset, the qualitative data in comparison to the quantitative data were analyzed to determine any patterns. There were significant, positive correlations between any positive or negative enactive mastery experience and all four factors of self-efficacy (0.39 to 0.50). This correlation further validates the instrument used to measure students' programming self-efficacy. Although both positive and negative experiences with coding were correlated to students having higher self-efficacy, persistence to retake the course could contribute to higher self-efficacy [7]. There were no correlations nor patterns identified relating any of the experiences mentioned and the growth mindset scores. Both the qualitative and quantitative analysis showed little connection between programming self-efficacy and mindset.

V. CONCLUSION

Although there was little connection between programming self-efficacy and mindset, there may be some potential explanations that point to a need for further research. The implemented self-efficacy instrument is a validated instrument for computer programming [6]. The implemented growth mindset instrument is validated [9], but not for domain-specific growth mindset. The sentence prefacing the survey prompting students to think specifically about computer programming may have been ineffective, so the results may reflect a general mindset. This could explain why there is no correlation between the students' mindset and self-efficacy scores. There should be further investigation into validating an instrument specific to computer programming mindset. This study supports the need for further research around domain-specific growth mindset [2]. Once a domain-specific growth mindset instrument is developed and validated, then the implementation of it should be compared to the implementation of the corresponding domain-specific self-efficacy instrument to investigate any potential relationship.

One interesting finding in comparing the quantitative and qualitative data was that some experience, regardless of good or bad, seemed to have a greater positive impact on programming self-efficacy than no experience. Bandura [7] discussed overcoming obstacles through persistence as enactive mastery experiences that build resilient sense of efficacy. The students that retake the course may have a higher self-efficacy because they are overcoming their past failures by attempting the course again. They may feel more capable and prepared because they have a better idea of what not to do compared to the students that have no experience – not know what to do be successful or not to do to avoid failure.

Based on the findings of this study, the two concepts that should be further researched are: domain-specific mindset and the impact of failure of a course on self-efficacy after the course is failed, the start of retaking the course, and after completing the course a second time.

ACKNOWLEDGMENT

This work was made possible by support from the Center for Teaching and Learning Excellence (CTLE) at Embry-Riddle Aeronautical University (ERAU). Any opinions, findings, and conclusions or recommendations expressed in this are those of the authors and do not necessarily reflect the views of ERAU or CTLE.

REFERENCES

[1] Snyder, K. E., Barr, S. M., Honken, N. B., Pittard, C. M., & Ralston, P. A. S. (2018). Navigating the first semester: An exploration of short-term changes in motivational beliefs among engineering undergraduates. *Journal of Engineering Education*, *107*(1), pp. 11–29.

[2] Scott, M. J., & Ghinea, G. (2014). On the domain-specificity of mindsets: The relationship between aptitude beliefs and programming practice. *IEEE Transactions on Education*, *57*(3), pp. 169–174.

[3] Hutchison, M. A., Follman, D. K., Sumpter, M., & Bodner, G. M. (2006). Factors influencing the self-efficacy beliefs of first-year engineerin students. *Journal of Engineering Education*, pp. 39–47.

[4] Bandura, A. (1994). Self-efficacy. In V.S. Ramachaudran (ed.), *Encyclopedia of human behavior* (Vol. 4, pp. 71-81). New York: Academic Press.

[5] Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84(2),* pp. 191–215.

[6] Ramalingam, and Siedenbeck. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. Journal of Educational Computing Research, 1998, 19(4), 367-381.

[7] Bandura, A. (1997). Self-efficacy: The exercise of control. New York, NY, US: W H Freeman/Times Books/ Henry Holt & Co.

[8] Dweck, C.S. (2007). Mindset: The New Psychology of Success, Ballantine Books.

[9] Devers, A. (2015). Thinking about intelligence: How students mindsets influence academic performance. *A Rising TIDE*, *8*(1). Retrieved from: http://www.smcm.edu/mat/wp-content/uploads/sites/73/2015/06/Allyson-Devers-2015.pdf.

[10] Dweck, C.S. (2008). Mindsets and math/science achievement. Paper prepared for the Carnegie Corporation of New York-Institute for Advanced Study Commission on Mathematics and Science Education. Retrieved from: https://www.growthmindsetmaths.com/uploads/2/3/7/7/23776169/mindset_and_math_science_achievement_-_nov_2013.pdf.

[11] Dweck, C.S. (1999). Self-theories: Their role in motivation, personality, and development. Philadelphia, PA: Psychology Press.

[12] Hatch, J. A. (2002). Doing qualitative research in education settings. Albany, NY: State University of New York Press.