ENCOURAGING A GROWTH MINDSET IN

COMPUTER PROGRAMMING INSTRUCTION FOR ADULT LEARNERS

Approved: ______________________________________ Date: ____April 14, 2017_______

keywords:
growth mindset
fixed mindset
computer programming
adult learners
community college

TABLE OF CONTENTS

PAGE

PAGE

ENCOURAGING A GROWTH MINDSET IN

COMPUTER PROGRAMMING INSTRUCTION FOR ADULT LEARNERS

A Seminar Paper

Presented to

The Graduate Faculty

University of Wisconsin-Platteville

In Partial Fulfillment of the

Requirement for the Degree

Masters of Science

in

Education

Adult Education

by

Paula A. Waite

2017

Abstract

While learning to program computers might be a popular idea, it can be a difficult skill to learn. Many students quickly excel, yet some struggle in computer programming courses. Many computer science instructors categorize students into two groups: those who seem to “get it” and those who don’t. It is interesting that researchers also categorized learners into two main groups: those with a *fixed mindset* and those with a *growth mindset*. Students with a fixed mindset believed intelligence and ability were innate and people could not alter them. At the other end of the spectrum, were students with a growth mindset, who believed effort and changes in strategies could improve intelligence and ability. A review of the research found that students with a growth mindset outperformed students with a fixed mindset in many disciplines, including computer programming courses. Research also showed that mindset became more fixed over the duration of programming courses. This indicated that there might be factors in computer programming courses that influenced mindset. Finally, techniques were identified which computer programming instructors might use to promote a growth mindset in their students.

Encouraging A Growth Mindset in Computer Programming Instruction for Adult Learners

Background

The Bureau of Labor Statistics (2015a) projected an increase in software developer employment at 17% and web developer employment at 24% from 2014 to 2024; both fields growing much faster than the average for all occupations. However, Gartner (as cited in Huntley and Young, 2015) projected that by 2020, universities would only produce enough graduates to fill 30% of the job openings. This projected shortage was concerning and many political, economic, and technical leaders took steps to encourage the population to take notice. Mark Surman, executive director of the Mozilla Foundation, for example, indicated that computer programming is the 4th literacy and everyone should know how the digital world works (Shein, 2014). In December 2013, U.S. President Barack Obama kicked off Computer Science Education Week by challenging young Americans to program their phones rather than simply playing with them (Finley, 2013). A year later, he again emphasized the importance of programming skills when he became the first president known to write a computer program (Beres, 2014).

While learning to write code might be a popular idea and lead to a lucrative career, it is no simple task. Studies showed that some students quickly excelled, yet many struggled in introductory computer programming courses (Jenkins, 2002). In fact, the pass rate in first semester computer science courses was consistent worldwide at 67.7% (Watson & Li, 2014). Perhaps even more worrisome: most entry-level web developers received their training in pursuit of an associate degree at two-year colleges where the 2010 completion rate over all disciplines was only 39% after six years (Bureau of Labor Statistics, U.S. Department of Labor, 2015b;

National Student Clearinghouse Research Center, 2017). Improving the completion rate of students in computer programming associate degree programs alone could help meet the projected computer programmer shortage in coming years.

Although many factors have contributed to the high failure rate described above, Carol Dweck (1999) showed that a learner's mindset towards one’s own learning had a crucial effect on his or her learning and success. She categorized learners into two groups: those with a *fixed mindset* and those with a *growth mindset* (Dweck, 1999). Students with a fixed mindset believed their intelligence and ability were innate and that they could not significantly change their own intelligence or learning ability (Dweck, 1999). Students with a growth mindset, on the other hand, believed that intelligence and ability improved with effort (Dweck, 1999). Individuals with a growth mindset tended to rise to challenges because they believed hard work and challenge would help them learn and grow. These students welcomed and persevered through difficult tasks (Dweck, 1999). Scott and Ghinea demonstrated in their study that students with a growth mindset outperformed students with a fixed mindset in computer programming classes (Scott & Ghinea, 2014). Further, Scott and Ghinea showed that without intervention, the mindsets of students became more fixed over the duration of programming courses attempted (Scott & Ghinea, 2014). This indicated there may be factors in computer science courses that affected mindset. Therefore, educators should consider mindset when designing and teaching computer programming courses and work to encourage a growth mindset, while discouraging a fixed mindset, in order to maximize student success. Ultimately, this may be the critical factor to improve completion rates in computer programming degree programs and to address the shortage of software developers in the United States.

Statement of the Problem

The purpose of this study is to determine if instructional techniques might be helpful to positively influence the mindset of adult learners in computer programming classes at community colleges. The goal would be to increase learner success by encouraging students to see their own intelligence and learning ability as something that can grow to meet educational occupational challenges.

Definition of Terms

Computer programming: also, referred to as programming, coding, web development, software development, software engineering or computer science. The practice and process of writing instructions that a computer can use to perform tasks or solve problems (Moursund, 2005). The field requires expertise in many areas, including analysis, design, algorithms, logic, problem solving, and testing (Moursund, 2005).

Fixed mindset: a belief that intelligence is a fixed entity and a person cannot significantly change it (Dweck, 1999).

Growth mindset: a belief that a person can expand his/her intelligence through effort (Dweck, 1999).

Mastery-oriented: behavior that focuses on the process (effort or strategy), especially when faced with a setback or difficulty (Kamins & Dweck, 1999).

Science, Technology, Engineering, and Math (STEM): the acronym, STEM, is widely used in education and economic discussions. Careers in those fields are the fastest growing, yet there is a shortage of skilled workers in those areas (Gonzalez & Kuenzi, 2012).

Self-theories: people's beliefs about themselves that lead them to think, feel and behave differently from others given the same situation (Dweck, 1999).

Purpose of the Study

The purpose of this study is to identify effective techniques for encouraging a growth mindset in computer programming instruction for adult learners in community colleges. This study provides a review of the literature to examine the role of mindset in learning computer programming and to overview instructional techniques that led to a growth mindset in students versus a fixed mindset. Finally, the study includes a model for computer programming instruction that might encourage a growth mindset and result in higher success rates for adult learners pursuing a career in computer programming.

Significance of the Study

Administrators and instructors can use this study to better understand factors in instruction that contribute to student success or failure in computer programming courses. Instructors can use the provided recommendations to improve their practice, thus increasing success and completion rates of their students. Fewer failures or withdrawals will result in an overall cost savings for students, who will not have to retake classes or choose other career paths. It will also mean increased income for colleges that receive funding based on completion rates. Finally, more students will realize their potential and enter the field as computer programmers while reducing the predicted shortage of information technology professionals in the United States.

Delimitations of Research

The Karmann Library at the University of Wisconsin – Platteville provided the references used for the review of literature for this study between September 2016 and February 2017.

Several online search engines provided by the EBSCO host were used. Key searches were: “growth mindset”, “mindset computer programming”, “community college persistence”, “growth mindset instructional techniques” and “attrition rates in computer programming courses”.

Method of Approach

The first review of literature was conducted on self-theory studies, specifically *fixed mindset* and *growth mindset*. A review of the research on the difficulties in learning programming followed. Next a review of research regarding persistence in community colleges and computer science programs, along with instructional techniques, was conducted. Then, a review was performed on mindset as it relates to the STEM disciplines and more specifically, computer programming. Finally, techniques that instructors can use to promote a growth mindset in their students and those techniques to avoid that might contribute to a fixed mindset were identified. In chapter 3 of this paper, the findings were summarized and recommendations made.

Chapter Two: Related Literature Review

Difficulties in Learning Programming

Learning to program computers is a complicated endeavor involving many areas of expertise. Unfortunately, many students learning to program computers encounter obstacles to success. This level of difficulty might be a factor in the 32.3% failure rate in first semester computer science courses reported over the last 30 years (Watson & Li, 2014). Without intervention, the prediction of a shortage in computer programmers in coming years will likely be accurate (Bureau of Labor Statistics, 2015a). In *Educating programmers: A reflection on barriers to deliberate practice*, researchers Scott and Ghinea (2013) argued that students needed to put in many practice hours (10,000) to become proficient at programming and suggested three main reasons why students did not practice enough.

First, the material might have been unlike anything they had previously encountered; learners needed to adapt to thinking about abstract concepts (Scott & Ghinea, 2013). To learn programming required both deep and surface learning which might have required the use of a variety of learning methods, something many students might not have done before (Scott & Ghinea, 2013). Beginning programming students often discovered that they could not learn the material using their typical study approaches (Scott & Ghinea, 2013). Bornat, Dehnadi, and Simon (2008) also highlighted the fact that computer programming was not widely taught prior to college as a significant factor that contributed to the difficulty in learning to program.

Second was the belief by students and teachers that some inherent aptitude was required to learn to program computers (Scott & Ghinea, 2013). While researchers attempted to test for programming aptitude, no studies have consistently and conclusively shown this to be true (Bornat, Dehnadi, & Simon, 2008). Yet, when many students encountered difficulty in learning computer programming material, many attributed that difficulty to a lack of aptitude (Scott & Ghinea, 2013).

Third was the fact that some students became frustrated or anxious by the challenge of learning something new and difficult (Scott & Ghinea, 2013). In computer programming, a single capital letter could be the difference between success (a running program) and failure (error messages). Errors in a program that prevented it from running derailed some students (Scott & Ghinea, 2013). In this case, the problem was identified as the students' lack of persistence to work through problems rather than other factors such as inadequate study skills (Jenkins, 2002).

Robins (2010) also set out to discover why many people found learning computer programming difficult. Robins reported that in addition to the high failure rate in computer programming courses, there was, historically, an unusually high rate of high grades, a few mid-range grades, and a high rate of low grades given for computer programming courses: a bimodal distribution. Patitsas, Berlin, Craig, and Easterbrook (2016) argued that while many computer science instructors reported a bimodal distribution of grades, some of those distributions might actually have been ambiguous distributions. Patitsas et al. (2016) showed that instructors who believed computer science grades were commonly thought to be bimodal, were more likely to label ambiguous distributions as bimodal.

Robins (2010) demonstrated that courses that covered highly dependent chunks of knowledge were more likely to result in bimodal grade distributions, while courses covering independent chunks of knowledge were more likely to result in a normal distribution. When chunks were dependent, then learning one chunk made learning a subsequent chunk easier;

failing to learn a chunk made learning the next chunk more difficult. Robins summarized this phenomenon as the *momentum effect*.

Traditionally, most programming courses were taught in strict, timed sequence without a mastery level contingency: concept A was taught, an assessment given, and regardless of the result of the assessment, concept B was taught, and so on. While this approach may work well to learn independent concepts, it may not be the best technique for disciplines involving dependent concepts. Robins (2010) explained that concepts in computer programming were very tightly connected: it was very difficult to understand one concept in programming independent of others. Robins compared learning programming to putting together a jigsaw puzzle: the puzzle became easier to assemble as more pieces fit together. Likewise, missing puzzle pieces made assembly difficult.   This tight integration of concepts meant some students found their previous learning strategies did not always work, and some instructors found that traditional teaching methods did not work either (Robins, 2010). Using his findings, Robins suggested a *mastery model approac*h to teaching and learning computer programming based on the theory that given time and the right learning conditions, anyone could learn to program. With a mastery model approach, students did not move on to learn concept B until they demonstrated mastery of concept A, regardless of how long that took.

Predictors of Success and Failure in Computer Programming

Concern around the high failure rate in computer programming courses resulted in many attempts to identify predictors of programming aptitude or success (Bornat, Dehnadi, & Simon, 2008). Bornat, Dehnadi, and Simon and Robins (2010) each presented a long list of studies and tests that evaluated numerous factors in the following categories: cognitive capacity, cognitive development, cognitive style, emotional health and well-being, and demographics. However, none of those factors consistently predicted success in learning programming (Bornat, Dehnadi, & Simon, 2008).

Katz, Allbritton, Arons, Wilson, and Soffa (2006) investigated why high-achieving students left computer science programs. They observed that more women who received a grade lower than a B (on a scale of A through F where A is the highest grade possible) left the program than men, even though women had outperformed men overall in the computer science courses. While the study was small, interviews with the women who left the program provided some insight into the attrition. For example, one student reported that she compared her own effort with that of her peers and concluded computer science was not for her, because she perceived that she put in more effort than some of her classmates (Katz et al., 2006). Another student set a goal of a B grade, but left the program after she scored a 75 on the first exam (Katz et al., 2006). The study did show a positive correlation between the number of calculus courses taken and the grades achieved in computer science courses (Katz et al., 2006).

Discoveries about Mindset

While research dispelled the effectiveness of tests to measure the ability to learn computer programming, there may be a valid predictor of student success, not just in programming, but in all disciplines. For well over thirty years, Carol Dweck (1999) researched *self-theories*, or people’s beliefs about themselves. Dweck (1999) discovered through numerous studies that populations tended to be split between two self-theories. One group believed the *incremental theory*: the self constantly changed in various increments. The other group believed the *entity theory*: the self was an unchanging entity. Dweck described individuals who believed in an incremental theory as having a *growth mindset*, while individuals who believed in the entity theory were categorized as having a *fixed mindset*.

Dweck researched academic, business and even athletic performance of those with a growth mindset versus a fixed mindset. Dweck (2006) discovered that people's self-theories could vary across domains, for example, intelligence, morality or athletic ability. An individual's theory of intelligence, or belief about whether intelligence was stable or malleable, might not be the same as his or her theory of athletic ability. Dweck's studies showed that people with a growth mindset generally outperformed those with a fixed mindset regardless of the task. She demonstrated that mindset was a critical contributor to success due to the difference in motivating factors for each mindset type (Dweck, 1999). It is interesting that the population distribution of fixed mindset (40%) and growth mindset (40%) was similar to the distribution of programming student grades described by Robins (2010) (Dweck, 1999).

Fixed mindset motivators.

Dweck (1999) discovered that individuals with a fixed mindset for intelligence had a tendency to focus on how much intelligence they possessed. Looking and feeling like they had high levels of intelligence were the top goals for people in this category (Dweck, 1999). They believed that if they accomplished a task without much effort, they were smart; if they had to work hard at a task, felt challenged, or even failed, then they felt inadequate or even "dumb" (Dweck, 1999). Dweck's studies showed that people with a fixed mindset tended to choose easier tasks over challenging tasks. Further, when faced with difficulty or failure, individuals with a fixed mindset tended to exhibit a helpless response: they blamed, disengaged, or quit (Dweck, 1999).

Growth mindset motivators.

Dweck (1999) discovered the factors that motivated individuals with a growth mindset for intelligence were very different from those with a fixed mindset. Individuals with a growth mindset tended to want to learn, as they believed learning would make them smarter (Dweck, 1999). They did not worry about looking smart or looking dumb, but instead looked for ways to challenge themselves (Dweck, 1999). They did not enjoy easy tasks, but instead appreciated effort, hard work, and learning (Dweck, 1999). When given a choice, those with a growth mindset regularly chose challenging tasks over easy tasks (Dweck, 1999). When faced with difficulty or failure, individuals with a growth mindset remained confident, self-motivated and they displayed grit and persistence (Dweck, 1999).

Study of mindset differences.

Blackwell, Trzesniewski, and Dweck's (2007) study of achievement as students transitioned to junior high school demonstrated the differences between the two mindsets. Blackwell et al. set out to answer these questions:

1. Were students' self-theories related to their levels of achievement?
2. Why was theory of intelligence related to grades?
3. Could theory of intelligence be changed to improve achievement?

Blackwell et al. selected mathematics as the focus subject due to its challenging nature, which they hypothesized would trigger behavior patterns related to theories of intelligence. Further, success in learning math, like computer programming, depended on successfully learning topics in a progression: missing an early concept made learning subsequent concepts more difficult, resulting in easily measurable achievement differences. The researchers scored students in their intelligence theories, learning goals and responses to failure (Blackwell et al., 2007). The results indicated that the students' theories of intelligence were significant predictors of math achievement (Blackwell et al., 2007). Students with a growth mindset for intelligence earned higher math grades than their peers with a fixed mindset: over two years, their math grades improved linearly, from a semester grade of 73 out of 100 in the fall of seventh grade, to 76 out of 100 in spring of eighth grade (Blackwell et al., 2007). The students with a fixed mindset, on the other hand, showed no improvement to a slight decrease (Blackwell et al., 2007).

Next, Blackwell et al. (2007) examined why the growth mindset students outperformed the fixed mindset group. Previous research demonstrated that students tended to have two different types of goals: learning goals, or goals focused on increasing skills and abilities, and performance goals, or goals focused on demonstrating competence (Dweck, 1999). Further, students tended to have two different responses to challenges or failures: a *helpless response*, in which the student gave up trying, or a *mastery response*, in which the student viewed the challenge as an opportunity to learn, remained positive, and persevered. In summary, the previous research identified the following factors related to students' mindset, goals and performance:

- students with a growth mindset tended to pursue learning goals versus performance goals;
- students with a growth mindset had positive thoughts about effort;
- students with learning goals demonstrated mastery responses in the face of challenge, while those with performance goals tended to exhibit a helpless response; and
- students with learning goals had better grades than students with performance goals (Dweck, 1999).

Blackwell et al. categorized the above into four factors for the study:

- learning goals,
- positive strategies,
- positive effort beliefs, and

- helpless responses to setbacks (Blackwell et al., 2007).

They conducted seven tests to determine the relationships of the factors; all seven tests showed that mediation was significant (Blackwell et al., 2007). This means the students with a growth mindset toward intelligence had stronger learning goals and were more likely to believe that hard work led to achievement than their fixed mindset counterparts (Blackwell et al., 2007). Further, students with learning goals and positive thoughts about effort responded to setbacks/failure with a change in strategy or with more effort (Blackwell et al., 2007). These students also had statistically significant higher grades than the students with a fixed mindset (Blackwell et al., 2007).

Blackwell et al. (2007) created a second part to the study to determine whether a person's theory of intelligence could be changed and whether that change resulted in improved performance. First, the researchers selected seventh graders in a school different from the first study. Again, Blackwell et al. analyzed math scores from sixth and seventh grade, but this time, they gave some students growth mindset training during the spring term of seventh grade. They taught selected students that people can grow their intelligence: the act of learning makes people smarter (Blackwell et al., 2007). Blackwell et al. also trained the control group, but the training covered memory and study skills rather than concepts regarding the development of a growth mindset. After the training, the testing indicated that the experimental group was significantly higher in growth mindset than the control group, demonstrating that mindset can be altered (Blackwell et al., 2007).

Blackwell et al. (2007) asked the math instructors of the students to report any changes they observed in their students' motivation. The teachers identified 27% of the experimental group with positive change, compared to 9% of the control group (Blackwell et al., 2007).

Teachers observed mastery responses: students asked for more help, worked over lunch periods and turned work in early to receive feedback before the assignment was due (Blackwell et al., 2007). The growth mindset group tended to use positive strategies to help them learn when faced with challenges.

Blackwell et al. (2007) also examined the mathematics achievement over time: called an *achievement trajectory*. As a whole, math achievement tended to decline when students transitioned to junior high; the first data points from sixth grade and first term of seventh grade supported this (Blackwell et al., 2007). In the second term of seventh grade, this trend continued in the control group; however, the group receiving growth mindset training reversed the declining trajectory and improved (Blackwell et al., 2007). Further, the study showed that students who had a growth mindset at the beginning of seventh grade improved the most after the training (Blackwell et al., 2007).

While Blackwell et al. (2007) conducted this study at only a couple schools, numerous researchers repeated similar studies at different schools (including colleges), with students of varying demographics, with similar results (Dweck, 2006; Scott & Ghinea, 2014; Yeager et al., 2016). The studies showed that

- individuals with a growth mindset outperformed individuals with a fixed mindset;
- students with a growth mindset welcomed challenge and persisted through difficulty; and
- mindset could be altered with training (Blackwell et al., 2007, Dweck, 2006; Scott & Ghinea, 2014; Yeager et al., 2016).

The Effect of Science, Technology, Engineering and Mathematics (STEM) Education on Mindset

Few studies investigated the relationship between programming and mindset, but several examined STEM and mindset (Blackwell et al., 2007; Cutts, Cutts, Draper, O’Donnell, &

Saffrey (2010); Dweck, 2006; Scott & Ghinea, 2014; Yeager et al., 2016). It was not surprising that the studies showed a growth mindset predicted higher success than a fixed mindset in the STEM disciplines (Blackwell et al., 2007; Cutts, et al., 2010; Dweck, 2006; Scott & Ghinea, 2014; Yeager et al., 2016). Perhaps not as expected, was the effect that STEM education had on mindset. In one study of first year engineering students, Reid and Ferguson (2014) demonstrated that students tended to drift away from a growth mindset toward a fixed mindset. In a 2014 study, Scott and Ghinea identified a similar trend. Because Scott and Ghinea's study explored the relationships between programming and mindset, it deserved a deeper exploration for this study. Scott and Ghinea (2014) set out to answer three questions:

- Did mindset for programming aptitude change differently from the mindset for general intelligence?
- Could students have a mindset for programming aptitude that was substantially different from their mindset for general intelligence?
- Did the mindset for programming aptitude better predict programming success compared to mindset for general intelligence?

The results showed that over the 8-week study, one third of the students became more fixed in their programming aptitude mindset (Scott & Ghinea, 2014). An interesting finding was that most students with fixed mindset for programming aptitude had a growth mindset for intelligence, indicating that students could have different mindsets across domains (Scott & Ghinea, 2014). The results also modestly showed that students do not associate their performance in computer programming with their overall sense of intelligence (Scott & Ghinea, 2014).

Techniques and Interventions

Findings discussed so far might be summarized in two key points:

- theories of intelligence, i.e., mindset could be changed, and
- a growth mindset was the preferred state because individuals with a growth mindset outperformed individuals with a fixed mindset.

While the Blackwell et al. (2007) studies clearly showed mindset training was one very effective technique to improve student success, it is important to explore and understand additional techniques and interventions that changed mindset positively, i.e., toward a growth mindset, and negatively, i.e., toward a fixed mindset. Kamins and Dweck (1999) and Mueller and Dweck (1998) demonstrated that feedback from adults influenced children's responses to later challenges. These studies involved participants performing a number of tasks during which they received one of three types of criticism or praise:

- person-oriented: comments on abilities, goodness, or worthiness,
- outcome-oriented: comments only on the result itself, i.e. correct or incorrect, or,
- process-oriented: comments on effort or strategy. (Kamins & Dweck, 1999; Mueller & Dweck, 1998).

Participants who received person-oriented feedback later displayed more fixed mindset, and helpless responses when faced with setbacks or challenges (Kamins & Dweck, 1999; Mueller & Dweck, 1998). Meanwhile, participants who received process-oriented feedback later displayed more mastery-oriented responses, such as using new strategies or more effort, when faced with setbacks or challenges (Kamins & Dweck, 1999; Muller & Dweck, 1998). Finally, those who received outcome-oriented feedback scored in the middle of the two extremes (Kamins & Dweck, 1999; Mueller & Dweck, 1998). These studies showed that to foster a growth mindset, educators should provide feedback on strategy and effort, and avoid feedback focused on the individual.

In a similar study, Rahman, Nandigam, and Tirumala (2015) found that using coaching techniques improved the pass rate and resulted in significant improvement in skill levels in their programming courses. In this study, the researchers' definition of coaching included an emphasis on helping students explore new ways to think and behave; encouraging learners to monitor their progress and analyze what worked and what didn't work (Rahman et al., 2015). That tactic was reminiscent of the strategy and effort feedback recommended in Kamins and Dweck's (1999) and Mueller and Dweck's (1998) studies. Rahman et al. also emphasized the importance of the educator mindset: teachers needed to believe that a student was capable of achieving their goals. In other words, educators also needed to have a growth mindset for their students. Their recommendations to instructors included:

- implement a learner-centered approach
- work to build relationships and community, and
- believe that students already have skills and knowledge and a strong desire to learn and achieve their goals (Rahman et al., 2015).

Murphy and Thomas (2008) and Rattan, Good, and Dweck (2012) supported a statement made in Rahman et al. (2015) that instructor mindset played an important role in student success. Rattan et al. found that educators with a fixed mindset for math intelligence were significantly more likely to:

- believe a student had low ability based on one poor performance,
- console the student for their lack of ability, and
- counsel the student out of the course.

Although these actions were well-intentioned, they may have contributed to the low numbers of students who pursued STEM fields (Rattan et al., 2012). In related work, Dweck (2006) showed that teachers with a fixed mindset ended up with different results from teachers with a growth mindset. Students in a class taught by an instructor with fixed mindset had the same ranking at the beginning and end of the semester, while all students in classes taught by an instructor with a growth mindset showed improvement.

Dweck (1999) also used her research to provide recommendations to encourage growth mindset via curriculum design and teaching techniques. For example, to reinforce a growth mindset in a student who easily completed a task, Dweck (1999) recommended the instructor apologize to the student for wasting his or her time, and find more challenging tasks or materials for the student. She cautioned that phrasing of the goals of an assignment, course or program can affect mindset of the students (Dweck, 1999). For example, an instructor might include verbiage in an assignment to acknowledge the challenge of the assignment and encourage practice with a newly introduced concept, thereby promoting a growth mindset. She also suggested that educators teach students how to learn from failure and embrace challenge (Dweck, 1999). Finally, Dweck (1999) advised teachers to set students up so they proved to themselves that by working harder, they performed better.

Cutts et al. (2010) utilized the aforementioned techniques to foster a growth mindset and affect performance in an introductory programming course. Specifically, the techniques were: mindset training, strategy and effort-based feedback, and a list of problem solving steps, i.e., strategies. Tutors conducted the mindset training intervention in four 10- to 15-minute sessions at the beginning of their small-group tutorial/lab sessions (Cutts et al., 2010). The strategy and effort-based feedback intervention was a script that was included with any evaluations shared with the student (Cutts et al., 2010). The script read:

> Remember, learning to program can take a surprising amount of time & effort – students may get there at different rates, but almost all students who put in the time & effort get there eventually. Making good use of the feedback on this sheet is an essential part of this process. (Cutts et al., 2010, p. 432)

The list of problem-solving steps, also known as the crib sheet intervention, included a collection of 35 methods successful students used to troubleshoot problems (Cutts et al., 2010). Both the control and test groups of students received this crib sheet. Tutors for the test group only used the crib sheet when helping students (Cutts et al., 2010). This approach reminded the students in the test group of the strategies they had to solve problems and modeled the technique for choosing the right strategy for the problem (Cutts et al., 2010). The study showed that students who received both the feedback script and the mindset training outperformed the control group (the mean for the group that received training was over 10% higher than the control) (Cutts et al., 2010).

Summary

Research showed that individuals with a growth mindset outperformed those with a fixed mindset. A number of studies demonstrated that students could have a mindset for a particular subject area such as computer programming or STEM courses that is different from their overall mindset, and that mindset could be changed. With nearly a 30% failure rate of first semester programming students and predicted shortage of programmers in the near future, instructors need to examine techniques to help students be successful with this difficult material. By creating curriculum and using techniques that transform fixed mindsets to growth mindsets, instructors can help improve completion and achievement rates in computer programming courses.

Chapter Three: Conclusions and Recommendations

This study provided a review of the literature about mindset and its role in learning computer programming. It also provided a discussion of several studies and findings related to methods to promote a growth mindset and minimize the opportunity for fixed mindset development. For reference, the appendix contains a straightforward list of recommendations and techniques for computer programming instructors to encourage growth mindsets in their students.

There were very limited studies available for review with regard to mindset and computer programming. Those referenced here used a relatively small sample size and were short in duration. The Cutts et al. (2010) study, for example, is a good candidate for repeated studies. It would also be interesting to measure instructor mindset for programming ability and the effect of that mindset on programming student mindset and success. Future studies could determine if the techniques recommended in the appendix prove to be effective in a large number of schools over longer duration. The results could help inform decisions about professional development for instructors and provide insight into curriculum improvements.

References

American Psychological Association. (2010). *Publication manual of the American Psychological Association* (6th ed.). Washington, DC: Author.

Beres, D. (2014, December 9). Obama writes his first line of code. The Huffington Post. Retrieved from http://www.huffingtonpost.com/2014/12/09/obama-code_n_6294036.html

Blackwell, L. S., Trzesniewski, K. H., & Dweck, C. S. (2007). Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child Development*, *78*(1), 246-263. doi:10.1111/j.1467-8624.2007.00995.x

Bornat, R., Dehnadi, S. and Simon (2008). Mental models, consistency and programming aptitude. *Tenth Australasian Computing Education Conference (2008), Wollongong, NSW, Australia. 78*, 53-62. Retrieved from http://crpit.com/confpapers/CRPITV78Bornat.pdf

Bureau of Labor Statistics, U.S. Department of Labor (2015a). *Occupational Outlook Handbook, 2016-17 Edition, Software Developers,* Retrieved from http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

Bureau of Labor Statistics, U.S. Department of Labor (2015b), *Occupational Outlook Handbook, 2016-17 Edition, Web Developers*, Retrieved from https://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm

Cutts, Q., Cutts, E., Draper, S., O’Donnell, P., & Saffrey, P. (2010). Manipulating mindset to positively influence introductory programming performance. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education – Special Interest Group on Computer Science Education '10.* doi:10.1145/1734263.1734409.

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

Dweck, C. S. (2006). *Mindset: The new psychology of success*. New York: Random House.

Finley, K. (2013, December 9). Obama says everyone should learn how to hack. *WIRED.* Retrieved from http://www.wired.com/2013/12/obama-code/

Gonzalez, H. B., & Kuenzi, J. J. (2012). *Science, technology, engineering, and mathematics (STEM) education: A primer.* Retrieved from https://fas.org/sgp/crs/misc/R42642.pdf

Huntley, H., & Young, A. (2015, April 21). *Service providers are waging war against U.S. talent shortage with unconventional methods*. Retrieved from Gartner website: https://www.gartner.com/doc/3034118/service-providers-waging-war-talent

Jenkins, T. (2002). On the difficulty of learning to program. *3rd Annual Conference Higher Education Academy Learning Teaching Support Network*, Loughborough, U.K. Retrieved from http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

Kamins, M. L., & Dweck, C. S. (1999). Person versus process praise and criticism: Implications for contingent self-worth and coping. *Developmental Psychology, 35(3),* 835-847. doi:10.1037//0012-1649.35.3.835

Katz, S., Allbritton, D., Aronis, J., Wilson, C., & Soffa, M. L. (2006). Gender, achievement, and persistence in an undergraduate computer science program. *Database for Advances in Information Systems, 37*(4). Retrieved from https://www.cs.virginia.edu/~soffa/publications/p42-katz.pdf.

Mueller, C. M., & Dweck, C. S. (1998). Praise for intelligence can undermine children's motivation and performance. *Journal of Personality and Social Psychology, 75(1),* 33-52. doi:10.1037//0022-3514.75.1.33

Moursund, D. G. (2005). *Increasing your expertise as a problem solver: some roles of computers.* Eugene, OR: International Society for Technology in Education. Retrieved from http://pages.uoregon.edu/moursund/Books/PS-Expertise/PS1996.pdf

Murphy, L., & Thomas, L. (2008). Dangers of a fixed mindset. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education '08.* doi:10.1145/1384271.1384344

National Student Clearinghouse Research Center. (2017). *The Role of Community Colleges in Postsecondary Success Community Colleges Outcomes Report*. Retrieved from https://studentclearinghouse.info/onestop/wp-content/uploads/Comm-Colleges-Outcomes-Report.pdf

Patitsas, E., Berlin, J., Craig, M., & Easterbrook, S. (2016). Evidence That Computer Science Grades Are Not Bimodal. *Proceedings of the 2016 Association for Computing Machinery Conference on International Computing Education Research '16.* doi:10.1145/2960310.2960312

Rahman, N., Nandigam D., & Tirumala, S.S. (2015). Teaching computer programming with a coaching mindset. *International Conference on E-Commerce Technology and E-Services, The 2nd World Congress on Computer Applications and Information Systems.* Retrieved from http://unitec.researchbank.ac.nz/handle/10652/3367

Rattan A., Good C., Dweck C. S. (2012). It's ok—Not everyone can be good at math: Instructors with an entity theory comfort (and demotivate) students. *Journal of Experimental Social Psychology, 48*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0022103111003027

Reid, K.J. & Ferguson, D. M. (2014). Do design experiences in engineering build a “growth mindset” in students? 2014 I*nstitute of Electrical and Electronics Engineers* Integrated STEM Education Conference, Princeton, NJ, 1-5. doi:10.1109/ISECon.2014.6891046

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, 20(1), pp. 37-71. doi:10.1080/08993401003612167

Scott, M. J., & Ghinea, G. (2013). Educating programmers: A reflection on barriers to deliberate practice. *The Higher Education Academy*. Retrieved from https://www.heacademy.ac.uk/system/files/cs_028_0.pdf

Scott, M. J., & Ghinea, G. (2014). On the domain-specificity of mindsets: The relationship between aptitude beliefs and programming practice. *Institute of Electrical and Electronics Engineers Transactions on Education, 57*(3), 169-174. doi:10.1109/te.2013.2288700

Shein, E. (2014). Should everybody learn to code? *Communications of the ACM*, *57*(2), 16-18. doi:10.1145/2557447

Watson, C., & Li, F. W. (2014, June). Failure rates in introductory programming revisited. *Proceedings of 2014 Conference on Innovation & Technology in Computer Science Education '14*. (pp. 39-44). doi:10.1145/2591708.2591749

Yeager, D. S., Romero, C., Paunesku, D., Hulleman, C. S., Schneider, B., Hinojosa, C., . . . Dweck, C. S. (2016). Using design thinking to improve psychological interventions: The case of the growth mindset during the transition to high school. *Journal of Educational Psychology*, *108*(3), 374–391. doi:10.1037/edu0000098

Appendix

List of Recommendations for Computer Programming Instructors

to Encourage a Growth Mindset

by

Paula A. Waite, pawaite@gmail.com

1. Understand self-theories: growth mindset and fixed mindset.
    a. Fixed mindset is a belief that abilities are innate or fixed.
    b. Growth mindset is a belief that abilities can be developed through effort and persistence.
    c. People with a growth mindset tend to outperform those with a fixed mindset, particularly when faced with challenges or difficulties.
    d. Mindset can be altered.
    e. People can have different mindsets for different disciplines.
2. Be able to infer whether a student is growth versus fixed mindset-oriented with regard to computer programming and intelligence in general.
3. Evaluate your own mindset regarding programming ability and cultivate a growth mindset if not already present. Students taught by instructors with growth mindsets tend to outperform students in courses taught by instructors with fixed mindsets.
4. Educate students in self-theories: growth mindset and fixed mindset. Consider using mindsetworks.com, for example.

5. Provide process-related feedback rather than person-oriented feedback. Help students attribute failure to factors under their control, for example, poor planning, lack of effort, ineffective strategies.
6. Present challenges, such as bugs or error messages, as opportunities to learn.
7. Use growth mindset messages. For example, demonstrate that people who make great contributions, the leaders in the field, are not born with special talents, but instead had to work to succeed. An internet search for “growth mindset messages for teachers” provides numerous phrases to help promote a growth mindset.
8. Set learning goals rather than performance goals.
9. Consider a mastery model approach to teaching and learning that ensures students master a concept before learning the next, related concept. This might mean giving credit for reworking problems a student got wrong.
10. Model life-long learning.