Investigating the Impact of an Incremental Mindset Intervention on Students' Beliefs and Programming Performance

by

Maria Jakeline de Guadalupe G. Celis Rangel

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Cognitive Science

Carleton University

Ottawa, ON

Abstract

Implicit theories of personal ability identify two kinds of mindsets, incremental and entity. Individuals with an incremental mindset believe in the value of effort, have mastery-oriented goals, and have been associated with higher academic performance. In contrast, entity individuals see effort as evidence of a lack of ability, and have been associated with lower academic performance. Prior research in implicit theories found that incremental beliefs can be taught via mindset interventions, and improvements in academic achievement were associated with these interventions, especially for in face of challenging subjects such as math or physics. Programing has also been identified as a challenging subject for students, but to date little research has been conducted to study the impact that implicit theories might have among novice programmers. To address this gap, this research tests the effect of an incremental mindset intervention on students' beliefs and programming performance in a controlled experiment, as well as analyzes factors related to the mindset construct specifically for the programming domain. Our key results show that as in prior work on general mindset (outside of programming), students who believe programming is a fixed ability do not believe in the value of investing effort. Moreover, compared to the control group, the incremental mindset intervention made participants significantly less entity oriented in their beliefs towards programming, and in turn they exhibited marginally more persistence in their programming activities, especially among those with less prior programming experience. However, this did not translate to improvements in performance on the programming activity.

Acknowledgements

I want to dedicate this thesis to the valuable people who hugely contributed on its completion. First and foremost, I would like to thank Kasia Mulder, Ph.D. for her incessant dedication and guidance throughout the process; for the enormous amounts of patience she always had and her valuable and tireless advice. Also thank you to all of the anonymous participants of the study, for dedicating their valuable time to the study and to the numerous pilots we ran; for their input and comments without which this project could have never reached a good port. Thank you also to the staff members of the Cognitive Science department for their tips and support, and to my library co-workers whose unwithering support and friendship helped me push through the hardest parts of the project. Also to my dear sister Ana whose experience and advice were of tremendous benefit to me; for her patience and diligence during our conversations, encouraging me to clear obstacles and carry on. To my mom for having always inspired me to grow and go beyond my comfort zone and to my dad who taught me to dream and work to reach my dreams.

Finally, to my husband Rafael, for being always by my side diligently supporting me, organizing and taking care of everything so I could be dedicated to my studies; for his wise and always practical advice. To Pablo, my oldest son, for being always open to listen my ideas and provide me with his points of views. And last but not least, to my youngest Adrian for staying always in good mood and ready to help.

Thank you God for giving me life, the ability to learn and the strength to keep going. Thank you all who supported me through every phase of the project, staying with me during the endless nights of reading, writing, thinking and talking, and to all those who took time from their activities to help me in the many things I interminably needed.

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter One

Introduction

Personal beliefs about the nature of ability set up cognitive frameworks that individuals use to interpret and react to their environment - these personal beliefs are also referred to as *implicit theories* because they are frequently not explicitly articulated by the person holding them. There are two key mindsets: entity and incremental. Individuals who hold the entity theory (or fixed mindset) believe that personal traits are fixed and cannot be enhanced. For these theorists, intelligence is an inborn trait, akin to, for instance, eye color. Individuals belonging to this group agree with statements such as: *"You have a certain amount of intelligence, and you really can't do much to change it"*. In contrast, individuals who subscribe to the incremental theory (or growth mindset) believe that personal traits such as intellect and skills are malleable and developed through practice and effort. These individuals have the belief that anyone can enhance their ability, if they receive the proper motivation, opportunity, and instruction. This second group of theorists agree with statements such as: *"You can always greatly change how intelligent you are"*.

Mindset theories (entity vs. incremental) have an important role in educational contexts, as they predict academic achievement. Specifically, a growth mindset leads students to engage in beneficial strategies and behaviors during learning activities (Aronson, Fried, & Good, 2002). Consequently, the sum of all these effects results in better academic performance (Blackwell, Trzesniewski, & Dweck, 2007; Dweck, 2006), even in the face of setbacks (Hong et al., 1999; Hu, Chen, & Tian, 2016). In particular, the impact of implicit theories is especially salient when students face transitions and challenges

(Dweck, 2006; Yeager, Johnson, Spitzer, Trzesniewski, Powers & Dweck, 2014), such as during academically challenging activities (Paunesku et al., 2015). One such activity is programming, where students struggle, for instance, when learning to program (Perkings, Hancock, Hobbs, Martin, & Simmons, 1986; Cutts, Cutts, Draper, O'Donnell & Saffrey 2010). Malleable views of programming ability might help students to achieve success. However, little research exists exploring the connection between students' mindset within the computer science domain and their achievement in programming activities, and the potential benefits of an incremental mindset intervention for students who are in the process of learning to program.

Dweck proposed that the two types of mindsets are evenly distributed among adults and children: About 40% of individuals hold an entity mindset, another 40% an incremental mindset, and 20% are not decided (Dweck & Master, 2005; Dweck & Molden, 2005; Dweck, 2011). Although mindset beliefs are relatively stable when left alone, these beliefs can be taught or induced via targeted interventions. For instance, implicit mindsets have been successfully manipulated, through extended workshops targeting the malleability of beliefs (Yeager, Trzesniewski, & Dweck, 2013; Blackwell et al., 2007) or short online sessions (Yeager et al., 2014; Paunesku et al., 2015). Growth mindset interventions have been successful in various settings for guiding students towards believing in their ability to learn, changing students' understanding of effort and failure, and helping them persevere in the face of difficulty.

Therefore, the primary goal of this work is to investigate the impact of an incremental mindset intervention in a controlled laboratory setting with an authentic instructional activity in the programming domain. We are primarily interested in whether the

intervention will impact participants' beliefs, making them to adopt more incremental beliefs, as well as improve in their actual performance on the programming activities. A secondary goal is to add to the research community's knowledge of the mindset construct specifically related to the programming domain, including the distribution of beliefs in this domain, the nature of the entity and incremental constructs, and the relationships between them.

This thesis will both build and extend on prior work. As far as the former, there is already established research showing that students' mindset can be manipulated and that doing so has beneficial academic outcomes (Blackwell et al., 2007; Paunesku et al., 2015; Yeager et al., 2016). Most of this work has been carried out in academic settings, i.e., in schools, where outcomes were measured in terms of students' GPAs. Much less work exists evaluating mindset in more controlled laboratory conditions, and to the best of our knowledge, none in the programming domain. Thus, our work will extend the studies by incorporating an authentic learning activity under laboratory conditions and measuring actual behaviors in addition to self-reports within the domain of computer programming.

Chapter Two

Related Work

2.1 Precursors of implicit theories

We begin with a brief background of research that lead to the development of implicit theories of personal abilities. One of the early precursors was the triadic reciprocity model developed by Bandura (1986). According to his model, each of its three components (person, behavior, and environment) influences and is influenced by the other two. Consequently, personal beliefs, which are an aspect of the "person" component, have an influence on an individual's behavior and vice-versa. Another precursor to development of implicit theories is seminal work by Diener and Dweck (1980), who studied children's responses after failure. Their work found that some children's responses exhibited a maladaptive helpless reaction when faced with a setback, which in turn hindered further progress. In contrast, other children exhibited an adaptive mastery reaction, such as evolving to use more sophisticated problem solving strategies. Importantly, Diener and Dweck (1980) found that maladaptive reactions were the result of attributions towards uncontrollable stable factors, whereas adaptive ones were the result of controllable and changeable factors. Some years later, Dweck and Leggett (1988) proposed an empirically-based model in which these adaptive and maladaptive behavioral patterns were the result of the implicit theories of personal intelligence and ability[1] that students held. This model

[1] The present work will refer to implicit theories of personal intelligence and ability also as *implicit theories* and/or *mindsets*.

can be identified as one of the milestones in the study of implicit theories. According to this model, people who subscribe to the incremental theory are more likely to adopt goals that focused on mastering the material (i.e., mastery goals), which in turn lead them to develop their abilities. When these individuals face failure, they interpret it as a normal consequence of learning and see it as a chance to improve their ability. In contrast, those who act in accordance to the entity theory tend to adopt goals that focus on demonstrating their existing competency (i.e., performance goals). When these individuals face setbacks, they attribute them to their lack of ability, which in turn results in aversion to the task and negative affect.

Even though implicit theories of personal abilities apply to multiple domains, such as weight management (Burnette, 2010), sports (Ommundsen, 2003), and leadership (Hoyt, Burnette, & Innella, 2012), the focus of this thesis is on the influence of personal beliefs in academic achievement and learning. We now describe related work in this area.

2.2 Impact of implicit theories

We have stated that implicit theories impact academic outcomes, but thus far have not described how this occurs. There is evidence that implicit theories have an effect on academic outcomes by influencing factors related to those outcomes (see Appendix A), as we now describe.

One influence of implicit theories is on the academic goals students adopt. Elliott and Dweck (1988) identified two main types of goals students can pursue: learning or performance. In the former, students set the goal to learn and measure their achievement against their own previous state. In the latter, individuals set the goal to demonstrate their

ability (e.g., want to demonstrate that their performance is superior to others). These two broad types of goals are related to students' implicit theories. Performance goals are more likely to be adopted by entity theorists (Blackwell et al., 2007; Dweck, 2006; Hong et al., 1999), because these individuals believe that ability is a fixed trait. These theorists want to demonstrate how much ability they have as a way to validate (in their mind) this fixed trait. In contrast, incremental theorists are more likely to adopt mastery goals, because for these individuals ability is malleable (a belief needed in order to adopt mastery goals, since these involve improving one's ability). The associations between implicit theories and goal orientation were first observed by Diener and Dweck (1980) and Dweck and Legget (1988) in studies with children, which were later confirmed by Blackwell et al. (2007). A recent meta-analysis conducted by Burnette, VanEpps, Boyle, Pollack and Finkel (2013) further supported that a student's mindset predicted the type of goal adopted.

Implicit theories also influence students' beliefs in the value of effort. Dweck (2006) observed that students with entity mindsets avoid effort at all costs because for them it suggests that they do not have the intelligence to perform the task *("If you have to strive, is because you are dumb or incompetent."*) and puts them at risk of having others realize that they are not intelligent. Moreover, if one lacks the ability to do something and one believes ability is fixed, investing effort will not help. In contrast, incremental theorists value effort as they believe it is needed to learn and increase their ability. Thus, incremental mindset students are more likely to achieve higher performance than entity mindset students, because they believe in the value of effort and are willing to invest it (Dweck, 2006).

Yet another factor related to implicit theories affecting academic outcomes are the attributions that students make about their failures or setbacks. Entity theorists attribute their failures to their lack of ability, whereas the incremental theorists attribute failure to lack of effort they put into the task (Dweck, 2006; Blackwell et al., 2007; Hong et al., 1999). Importantly, entity theorists have the belief that they do not have control over a failure they experienced (because they can't change the amount of ability they already have). On the other hand incremental theorists believe they do have control and that by deploying more effort towards the task, they can improve. (Dweck, 2006). Blackwell et al. (2007) confirmed these findings, showing that individuals with an entity mindset were more likely to agree with statements such as *"I am just not good at this subject"* or *"I really didn't like the subject"*. In contrast, incremental mindset individuals agreed more with statements such as *"I didn´t study hard enough"* or *"I didn't go about studying in the right way"*. To assess the impact of these beliefs on academic outcomes, Blackwell et al. (2007) followed students' GPAs over two years. Their results showed incremental theorists' grades increased over time, while entity theorists' grades remained stable.

The strategies students adopt are also influenced by implicit theories (Blackwell et al., 2007), which in turn influence academic achievement. Entity theorists are more likely to follow ineffective strategies such as limited persistence in the face of difficulty, or rely on a single study strategy even if it is not useful. Nussbaum and Dweck (2008) found that another detrimental strategy used by entity theorists is to compare themselves to students who performed even worse than them, as a way to minimize their own poor results. In contrast, incremental theorists are more likely to rely on effective strategies, such as to

escalate effort in the face of setbacks, or to take remedial courses related to their failure (Robins & Pals, 2002).

Another consequence of the type of implicit theory a student holds is affect. King, McInerney, and Watkins (2012) found that students who held an entity theory were more likely to experience negative affect during school activities (e.g., anxiety, shame, anger, and boredom). Luo, Lee, Ng, and Ong (2014) explored the relationships among math achievement, affect, and implicit theories. Participants were Singapore 8th graders, who were followed during the school year via online surveys. By implementing structural equation modeling, researchers found that incremental mindsets positively correlated with positive emotions (such as enjoyment and pride), which resulted in higher achievement in comparison to entity theorists.

Now that we have outlined the key factors that influence academic achievement, including goal orientations, effort beliefs, attributions regarding failure, strategies, and affect, we briefly describe the work integrating these factors into a single model, subsequently used to describe how they influence academic achievement. While a number of different models have been proposed (Hong et al., 1999; Blackwell et al., 2007; Robins & Pals, 2002) overall, these models are in agreement that implicit theories impact the factors discussed above and in turn academic achievement. However, the exact details of the specific relationships among these components are still under study. For instance, one of the open questions regards the precise nature of the meditational role of the goal construct. Robins and Pals (2002) used self-reported information as input to a path analysis and found that implicit theories most strongly predict academic goals, which then predict the four mediational factors. In contrast, Hong et al.'s (1999) model indicates that implicit

theories directly predict all factors (i.e., goals, strategies, and other variables are all on the same "level" of the model).

2.3 Incremental mindset interventions can improve academic performance

In addition to identifying factors predicted by implicit theories and building theoretical models based on these, another strand of work has focused on exploring the value of manipulating student mindset to encourage an incremental one (as this is the mindset needed for academic success). These mindset interventions have been implemented in various ways, including those administered through extended workshops lasting several weeks (Blackwell et al., 2007), surveys (Zhao, Zhang, & Vance, 2013), peer mentoring (Aronson et al., 2002), and more recently in shorter versions delivered via computer or paper and pencil (Yeager et al., 2014; Paunesku et al., 2015). The interventions typically include three key aspects. The first is to provide the participants with scientific evidence, which conveys the message regarding the malleability of personal intelligence and abilities. The second aspect is to ask participants to express their own interpretation of what they learned from reading the scientific evidence and to identify the personal relevance that this information might have for them. This is important because it helps participants to internalize and adopt the mindset through the "saying-is-believing effect". The third aspect consists of writing a letter to another student who is facing similar struggles about how malleability of intelligence and related concepts can help them to succeed.

Interventions aimed at encouraging students to adopt an incremental mindset have been successfully implemented (Aronson et al., 2002; Blackwell et al., 2007; Paunesku et

al., 2015; Yeager et al., 2016; Hu et al., 2016), where success was measured by assessing students' shift in mindset and/or with respect to academic achievements (e.g., GPA's or Standard Tests Scores). This is critical as it confirms that growth mindsets can be taught and, as a consequence, improvement in academic achievement can be reached. To illustrate one of these studies, Blackwell et al. (2007, Study 2) implemented an incremental mindset intervention to teach 7th graders about the malleability of their abilities via eight in-school sessions. The intervention emphasized that every time students invested effort to learn something new, their brain generated new connections, leading to increased ability. Blackwell et al. (2007) compared the effects of this type of intervention (experimental group) against a control group who was only taught about study skills. Their results showed that, over a two year period, the control group declined in their math grades, while the incremental mindset intervention group maintained and even improved their grades.

Empirical evidence shows that mindset interventions are especially beneficial in the face of various challenges, with their effects diminished in situations of success (Dweck, 2006). These challenges can take a variety of forms. For instance, they can be related to a certain period in a student's life, such as during the adolescence or during the transition to college or university (Good, Aronson, & Inzlicht, (2003); Blackwell et al., 2007; Yeager et al., 2016). Yeager et al. (2014) implemented an incremental mindset manipulation among 9th grade students who were in a transitional period and were struggling with adjusting to this transition. The results showed that in addition to an improvement in academic achievement, maladaptive social behaviors such as aggression and social exclusion decreased among the experimental group in comparison to the control. As a second example, students who have historically been more likely to abandon their studies,

such as those belonging to minority groups (e.g., African-American, Hispanic or females in male dominated fields) may especially benefit from mindset interventions (Good et al., 2003). Aronson et al. (2002) implemented an incremental intervention through three one-hour sessions among African-American & Caucasian undergraduate students to manipulate their beliefs towards incremental mindsets. Even though this intervention focused on the malleability of personality traits (rather than academic abilities), the intervention did increase students' academic outcomes and enjoyment of the class material in comparison to the control condition.

In general, mindset interventions result in larger improvements in academic achievement in challenging domains. Paunesku et al. (2015) assessed the impact of a short 45 minute online incremental mindset intervention in four core courses (Math, English, Science and Social Sciences) among 9^{th} to 12^{th} graders. Their findings were clear: Satisfactory course completion was higher in math (perceived as hard by students) than in social studies (perceived by students as not that difficult – almost no gains in this area). In another study conducted by Good et al. (2003) similar results were obtained (larger gains for math and smaller gains for reading).

Yet another example of a challenge-related situation pertains to instances involving students being exposed to situations where they may initially fail or face of setbacks such as negative performance feedback. To illustrate, Hong et al. (1999, Study 3) investigated the impact of a mindset intervention on Chinese undergraduate students. The goal was to investigate reactions of students following either negative or positive performance feedback in each condition (incremental vs. entity). Task complexity was set to be high in order to make it difficult for students to evaluate their performance – this was needed

because the feedback provided was fictitious. Specifically, once participants finished the task, half of them were given fictitious positive feedback and the other half fictitious negative feedback. The target dependent variable was self-reported likelihood to take remedial actions after success or failure feedback. Students in the entity mindset group who received negative feedback were less likely to report willingness to take remedial courses, as compared to the incremental mindset group. Additionally, with regards to the attributions about effort, students manipulated with entity mindset attributed their poor performance to their ability. In contrast, students manipulated with incremental mindsets attributed their poor performance to (almost equally) to their ability and effort invested behind the task. In a more recent experiment, Hu et al. (2016) demonstrated that an implicit mindset intervention mitigated the adverse effects of negative performance feedback by reducing negative emotions.

2.4 Incremental mindset interventions in computer science

The empirical work reviewed above described how individual beliefs in one's ability plays a role in academic success or failure. The goal of this section is to describe work on mindset in the context of the programming domain.

Above, we stated that mindset interventions are the most effective for challenging domains and programming certainly falls into this category. Specifically, learning to program is not an easy task, one that many students struggle with, resulting in low academic

performance (Guzdial 2011). This is especially problematic because computer science is a continuously growing field[2].

To tackle this issue, computer education research has been mainly focused on the study of different ways in which programming can be taught, with a focus on programming methods rather students' mindsets (Chookittikul & Chookittikul, 2008; Siddiqi, Harrison & Siddiqi, 2010). There has also been some work on the development of new and engaging programming tools, such as games (Tillmann & Xie, 2011; Doerschuk, Juarez, Liu, Vincent, Doss & Mann, 2013). According to Jenkins (2002), "if students struggle to learn something (in computer programming), it follows that this thing is for some reason difficult to learn." (p. 1). However, to date, little research has focused on the understanding individual differences related to programming, and particularly ones related to mindset. One of the few examples is the work from Rogerson (2010). Her qualitative research found that the attributions students make about their successes and failures in coding are critical factors influencing outcomes. Although Rogerson does not explicitly associate her results with implicit theories, there appears to be a connection. For instance, she reported that some participants attributed their success to their own ability ("*Wow it worked! I did that! I made it work!" p. 159),* which is aligned with individuals with incremental mindsets. In contrast, other students attributed their success to external incontrollable factors, such as good luck (*"See you get to a point where you're so used to failing at something, even when you realize I did pretty good at that point, it's really hard to even acknowledge because you're thinking,*

[2] http://brookfieldinstitute.ca/wp-content/uploads/2016/07/The-State-of-Canadas-Tech-Sector-2016.pdf ; p. 39 and 40

ok that was probably a lucky break type set up.") – this type of attribution is reminiscent to ones made by individuals with entity mindsets.

Key findings from Kinnunen (2012) support the dichotomized view implied in Rogerson's research. Kinnunen reported that "some students reflect negative views of their efficacy, even after having a positive programming experience" whereas "other students in contrast, after negative programming experiences still have a positive outlook on their efficacy." (p. 1). Simon (2008) also observed that students in programming courses presented these same dichotomized reactions to challenge and failure.

To the best of our knowledge, only a few studies have been conducted that investigate the impact of incremental interventions have on helping students in the process of programming education. Simon et al. (2008) implemented a mindset intervention among undergrad students in the computer science program. Specifically, the researchers presented an in class lecture to communicate the concepts of implicit theories, followed by a "saying-is-believing" exercise from Aronson et al. (2002); students' mindsets were assessed before and after the intervention. Their results showed that the intervention did not change participants' beliefs to make them more incremental oriented. Cutts et al. (2010) also explored the potential of mindset interventions on programming grades by exposing (presumably college) participants to different interventions across the academic year – we focus on the most relevant ones here. One of these was the mindset training intervention, which was implemented via four sequential 10 to 15 minutes presentations, at the beginning of a two hour laboratory session, where tutors explained the concepts related to implicit theories. A second intervention, the crib-sheet intervention, involved giving students a crib sheet containing a list of suggestions to try when stuck. Finally, the rubric

intervention involved including an incremental message with the feedback given on the biweekly assignment (the assignment feedback was given to all students). The goal behind the message was to remind participants about their ability to overcome challenges. The mindset training intervention alone (e.g., without the rubric) changed participants' beliefs but did not improve programming grades. In contrast, the intervention that showed a positive impact on performance was the condition where participants were exposed to the mindset training and the rubric intervention (without the crib-sheet).

2.5 Summary

In summary, two key findings arise. First, when students hold an incremental mindset, they will be more likely to engage in beneficial behaviors (e.g., deploy more effort (in comparison to their entity peers) to progress in their skills via practice and hard work, value effort, believing that poor results/setbacks are the consequence of little effort and not to the lack of ability). Second, as a consequence of these behavioral patterns, improvements in academic achievement are obtained. The second key finding is that mindset beliefs can be manipulated by interventions and that doing so can improve academic outcomes.

To date, the majority of work has investigated mindset in school settings. While involving authentic contexts is certainly important, it means that the impact of a mindset intervention cannot be assessed in the context of a specific instructional activity and behaviors related to that activity (i.e., because doing so would not be practical). The laboratory studies that have been conducted manipulating mindset have not included authentic learning situations and in the majority of cases have relied on self-reports of behaviors rather than assessed outcomes through less subjective measures. Moreover, it

should be noted that little research has been conducted to investigate the impact of mindset interventions in the programming domain, or the relationships between the related constructs in that domain. To address these gaps, we now describe our research.

Chapter 3

Phase One: Online Survey

Before we could conduct a laboratory study to test the impact of a mindset intervention, we had to find suitable participants. Accordingly, the goals of phase one were to screen and recruit the target population of participants for phase two (the laboratory study), as well as to add to the understanding of the mindset construct by addressing the following research questions:

(1) Is the distribution of entity and incremental theorists (i.e., individuals who are classified as entity or as incremental) consistent with previous research? While various sources commonly advocate the 40% entity / 40% incremental / 20% in between rule (Dweck & Master, 2005; Dweck & Molden, 2005; Dweck, 2011), this appears to be primarily a theoretical rather than empirical proposal. To the best of our knowledge, there is very little data available, and even less regarding programming beliefs.

(2) Should the entity and incremental constructs be treated as separate unipolar constructs or can they be reduced to a single bipolar mindset construct? Initially, mindset beliefs were analyzed as a unipolar entity construct (Hong et al., 1999) but later, Blackwell et al. (2007) handled them as a continuum (bipolar construct). Since this question impacts the analysis we plan to conduct in phase 2, answering it is important, both for beliefs in general ability and programming-specific ability beliefs.

(3) What are the relationships among the target constructs? Answering this question will shed light on whether our results in the programming domain mirror prior findings on the belief in ability in other domains, as well as the belief in general, domain-independent ability.

To answer these questions, this phase consisted of an online survey that collected demographic information, participants' experience in the programming domain, and information about mindset and related constructs. To interpret effect sizes related to the below reported Pearson correlations, we use the convention from Evans (1996), where in absolute terms, .00 to .19 is considered "very weak", .20 to .39 is considered "weak", .40 to .59 is considered "moderate", .60 to .79 is considered "strong", and .80 to 1 is considered "very strong".

3.1 Method

3.1.1 Participants and recruitment

Participants were undergrad students from Carleton University. They were recruited through SONA, posters, and announcements in Carleton classes. Those recruited via SONA, regardless of whether they signed up for the laboratory session, obtained .25% extra credit as a token of appreciation for their participation. Participants recruited externally from SONA did not receive any compensation. We received research ethics board clearance for both study phases (#105858; November 22, 2016). The recruitment period lasted about two and a half months (SONA: from January 25th to April 7th, 2017; outside of SONA: from February 2nd to April 16th, 2017). A total of 436 participants (269 female) completed the survey (226 recruited from SONA). All participants were above 18

years old (M = 21.08; SD = 4.10); nearly 40% were in their first year (39.40%), another 20.90% were in their second year, and the rest were evenly distributed among third and fourth year. In terms of ethnicity, a little over half of the participants (54.80%) were Caucasian/European, 10.60% South Asian (e.g., Indian), 9.40% were Black (e.g., African American), and the rest were distributed among Arab (e.g., Saudi), East Asian (e.g., Chinese), Southeast and West Asian (e.g., Filipino and Afghan respectively). Participants were varied with respect to their major: 16.10% were in Psychology and the rest were spread among 33 different programs and majors (such as Cognitive Science, Computer Science, Criminology, Engineering and Communications).

3.1.2 Materials and Measures

To obtain data on ability and related aspects, we used the "Online Survey" (see Appendix B). It consisted of three questionnaires: (1) demographics, (2) prior computer programming experience, and (3) the Student Questionnaire.

Demographics. Seven questions asked about participants' general demographic characteristics such as age, gender, year of study, ethnicity, major, and first language.

Prior programming experience. Three questions asked participants about their previous experience in (1) programming, (2) using visual programming tools and (3) prior used of Snap! The first question was used to differentiate between experienced vs. non-experienced programmers, needed for screening of participants for phase two.

Student Questionnaire. This questionnaire integrated different instruments that were used to measure several motivational constructs, including (1) general implicit theories of ability (i.e., mindsets), (2) customized implicit theories of ability for the programming domain,

(3) effort beliefs, (4) achievement goals, (5) helpless attributions and strategies in response to failure[3]. Unless otherwise stated, all questionnaires used a six-point Likert-type scale from 1 (agree strongly) to 6 (disagree strongly). To assess the instrument reliability, Cronbach's alfa was computed and unidimensional scales convention was applied, where values below 0.50 are considered unacceptable, and reasonably good values are those between .65 and .80.

General implicit theories of ability. To measure general belief in ability (general mindset), we used the established mindset instrument (1997; Dweck, 1999). The questionnaire consisted of eight items, four entity theory items (e.g., "You have a certain amount of intelligence, and you really can't do much to change it"), and four incremental theory items (e.g., "You can always greatly change how intelligent you are"). For both entity and incremental items, the lower the score obtained, the higher the endorsement of the corresponding belief in ability, with negative beliefs in the entity scale (i.e., negative because they are harmful to learning) and with positive beliefs in the incremental scale. (i.e., positive because they are conducive to learning). The internal reliability of the entity measure was Cronbach's $\alpha = .87$ ($N = 436$), $M = 4.03$, and $SD = 1.04$; for the incremental measure Cronbach's $\alpha = .89$, $N = 426$, $M = 2.74$, and $SD = .97$.

Customized implicit theories of ability for the programming domain. To measure participants' belief in their ability in the programming domain (programming mindset), we

[3] The questionnaire also included six additional items related to programming. We initially created these for exploratory purposes, but after consideration, decided to not include them in the final analysis due to the lack of previous evidence regarding the validity of these items.

took all eight items from the general mindset instrument and adapted the phrasing to customize each one to be specific for the programming domain. This approach was used by other scholars to adapt the general mindset belief questionnaire to specific target domains, like leadership (Aronson et al., 2002; Castella & Byrne, 2015; Burnette, 2010; Hoyt et al., 2012). Therefore, there were four entity theory items related to programming (e.g., "You have a certain amount of computer programming ability, and you really can't do much to change it"), and four incremental theory items about programing (e.g., "You can always substantially change how much computer programming ability you have"). For the sake of consistency with the general mindset instrument, for both the entity and incremental items, the lower the score obtained, the higher the endorsement of those programming beliefs. The internal reliability of the entity measure was Cronbach's $\alpha = .85$ ($N = 436$), $M = 4.41$, and $SD = .91$; for the incremental measure Cronbach's $\alpha = .83$, $N = 426$, $M = 2.42$, and $SD = .80$.

Effort beliefs. To measure participants' belief in effort, we used the instrument adopted by Blackwell et al. (2007). This questionnaire contained five items associated with the general entity mindset, and four items associated with the general incremental mindset (this association was established by prior work (Sorich & Dweck, 1996)). Items associated with entity mindset measured the belief that effort has a negative relation with ability ("To tell the truth, when I work hard at my schoolwork, it makes me feel like I'm not very smart), and that it is ineffective in achieving positive outcomes ("If you're not good at a subject, working hard won't make you good at it"). Effort items associated with incremental mindset measured the belief that effort leads to positive outcomes (e.g., "If an assignment is hard, it means I'll probably learn a lot doing it"). For both sets of items (entity and

incremental), the lower the score the higher the endorsement of those effort beliefs. Entity measure Cronbach's α = .75, *N* = 436, *M* = 4.27, *SD* = .86; Incremental measure Cronbach's α= .57 (which was the lowest Cronbach's α found, but still moderate, and thus deemed acceptable for the present analysis) *N* = 426, *M* = 2.44, *SD* = .72.

Achievement goals. We used the same questions as ones used by Blackwell et al. (2007), with items assessing each of the three goal orientations, namely learning or mastery goals, performance-avoidance-goals, and performance-approach goals. These items were based on the "Manual for the Patterns of Adaptive Learning Scales" (PALS 2002, for the learning and performance avoidance goals). The learning goal construct consisted of three items which measured the value of learning ("An important reason why I do my school work is because I like to learn new things"). The lower the score obtained, the higher the endorsement of learning goal-related beliefs. This construct has been associated with incremental beliefs in prior work (Blackwell et al., 2007). (Cronbach's α = .72, *N* = 436, *M* = 2.74, *SD* = .90). The three items used to measure performance-avoidance goals measured how important one considers to not be seen as unintelligent by peers ("It's very important to me that I don't look stupid in class"). A low score indicates a high endorsement of the performance approach beliefs. (Cronbach's α = .80, *N* = 436, *M* = 3.71, *SD* = 1.14). Finally, the performance-approach goals assess how highly one values good performance, particularly as a way of demonstrating ability ("The main thing I want when I do my school work is to show how good I am at it"). A low score indicates a high endorsement of the performance approach beliefs. (Cronbach's α = .65, *N* = 436, *M* = 2.84, *SD* = .87).

Helpless attributions and strategies. To assess participants' self-reported response patterns (helpless-oriented attributions and positive strategies) in the face of academic

struggle, we borrowed the hypothetical failure scenario used by Blackwell et al. (2007). This fictitious scenario puts the students in a situation where they get an unexpectedly poor grade on a test, and asks them to report on what they would do in such a situation. The scenario is the following:

> *Instructions: When you read this story, pretend that it really happened to you and try to picture how you would feel and what you would do if it happened:*
>
> *You start a new class at the beginning of the year and you really like the subject and the teacher. You think you know the subject pretty -well, so you study a medium amount for the first quiz. When you take the quiz, you think you did okay, even though there were some questions you didn't know the answer for. Then the class gets their quizzes back and you find out your score: you only got a 54, and that's an F.*

To assess helpless-oriented attributions, participants were asked to respond how much they think their ability or other factors caused the failure. For this purpose, four helpless-oriented items were included (e.g., "I wasn't smart enough") and rated using a 1 to 6 point rating scale, where 1 was "very true" and 6 was "not at all true". Consequently, lower scores indicated higher endorsement of helpless-oriented attributions. (Cronbach's $\alpha = .67$, $N = 436$, $M = 4.10$, $SD = 1.10$). To assess positive and negative strategies, two positive, effort-based strategies (e.g., "I would work harder in this class from now on") and two negative, effort-avoidant strategies (e.g., "I would try not to take this subject ever again") were included with the same scale used to grade helpless-oriented attributions. The lower the score the higher the endorsement with those strategies. (Positive strategies Cronbach's $\alpha = .67$, $N = 436$, $M = 4.63$, $SD = 1.25$; Negative strategies Cronbach's $\alpha = .84$, $N = 436$, $M = 1.60$, $SD = .85$).

3.1.3 Procedure

Phase one was done online using the Qualtrics survey software. Once participants were recruited via any of the described methods, they used the provided link to access the survey. As soon as they finished filling in the study questionnaires, Qualtrics automatically sent participants one of the following two e-mails. The first was a "thank you e-mail", sent to participants who did not qualify for the laboratory session (details below), thanking them for participation (see Appendix C for the e-mail text). The second e-mail was sent to participants who did qualify for the laboratory session, and so included an invitation to participate in the laboratory session and instructions on how to sign up for it (see Appendix D).

Criteria used to screen participants for the laboratory session. To identify which participants were eligible to participate in the laboratory session, we used two criteria: (1) participants needed to have little or no experience with programming, necessary to avoid any potential ceiling effects in the programming activities, and (2) participants needed to endorse an entity mindset, at least to some degree – this was needed since the planned incremental mindset intervention would not have an effect on individuals who already endorsed an incremental mindset.

To identify participants with little or no programming experience, we relied on answers to the "prior programming experience" questionnaire. We focused on the answer to the question "How much experience do you have with computer programming?" as it superseded the other two questions related to visual programming languages. Students whose responses were "None" and "Some" were candidates for the laboratory session, but only if they fulfilled the second criteria related to mindset. To identify participants with

entity mindsets (i.e., ones who were not at ceiling in terms of their incremental beliefs), we computed a mean score for three of the entity programming items[4] (e.g., "You have a certain amount of computer programming ability, and you really can't do much to change it"), and established a threshold level of four (in the six-point scale where 1 was "Agree strongly" and 6 was "Disagree strongly"; more entity endorsement was represented by lower scores). Participants scoring below or at this threshold of four were invited to participate in the laboratory session.

This screening procedure was based on prior work from Hong et al. (1999), Erdley, Cain, Loomis, Dumas-Hines, & Dweck, (1997)., Blackwell et al. (2007), and Aronson et al. (2002). Hong et al. (1999) classified participants as having an entity mindset if the average score of the three general belief entity items were below or equal to three. Erdley et al. (1997) also used a cut off to classify individuals as having an entity mindset (or not), but the threshold they used was the mean score of the entity items (M = 3.39, Study 1), with participants below this threshold assigned to the entity group. In contrast, Blackwell et al. (2007) reverse scored the incremental mindset items and calculated an overall mean of all items, where low scores indicated entity endorsers and high scores incremental endorsers. To identify the threshold distinguishing these two groups, Blackwell et al. (2007) added or subtracted one standard deviation (SD = .97) to the implicit theory mean score (M = 4.50) - participants below the first value (3.50) were considered as entity

[4] Hong et al. (1999) computed the mean score to categorize participants as entity or incremental mindsets using three entity items. Inspired by this method, we also computed the mean score for the screening using three entity items. However, for the rest of the analysis we used four items.

oriented whereas those above the second value (5.40) were considered incremental (participants who scored in the middle were considered to have mixed beliefs). While Aronson et al. (2002) did not explicitly set up a threshold level to differentiate between the two mindset groups, they refer to participants with more (or less) entity and incremental beliefs based on a median split of an overall collapsed implicit theory score. In sum, based on these various prior works we chose a threshold of four, and selected as candidate participants those who scored below this threshold on the entity items.

3.2 Results

We now present the results related to our three research questions about the mindset construct.

3.2.1 Distribution of beliefs

Prior work from Dweck (Dweck & Master, 2005; Dweck & Molden, 2005; Dweck, 2011) reports that belief distributions follow the 40% - 20% - 40% pattern, for entity, mixed and incremental theorists, respectively (note that this means the mindset construct is trichotomized, with people assigned to a given category based on their answers to the mindset questionnaire). However, to the best of our knowledge, this claim is not empirically based, nor is the method used for classifying individuals as entity or incremental clear. To obtain information on belief distributions, we used the method in Blackwell et al. (2007), who did classify individuals as either entity or incremental (although did not explicitly compute and report distributions). This method consists of reverse scoring the incremental items, and obtaining a grand mean of all the items (entity and incremental). Figures 3.1 and 3.2 present the distributions we obtained using this

method, for general ability beliefs and for programming beliefs, respectively. Participants who fall one standard deviation below or above the grand mean are classified as entity and incremental, respectively – the results related to the percentage of participants falling into each of these three categories are in Table 3.1. The results for general mindset and programming-specific mindset are similar, and neither are consistent with Dweck's reports (Dweck & Master, 2005; Dweck & Molden, 2005; Dweck, 2011) but are consistent with findings from Tempelaar, Rienties, Giesbers, and Gijselaers (2015), where about two thirds of the participants reported beliefs that were not strongly entity or incremental and the rest were evenly distributed between entity and incremental mindsets (see Table 3.1). Thus, overall, we did not find evidence for the 40-20-40 distribution advocated by Dweck.

Table 3.1
Percentage of entity, incremental, and "mixed" mindsets for general beliefs in ability and programming beliefs in comparison to prior results from Tempelaar et al. (2015)

Belief	General ability beliefs	Programming beliefs	Tempelaar general beliefs
Entity	16	19	18
Mixed	65	65	64
Incremental	19	16	18

Note. All data presented in percentages.

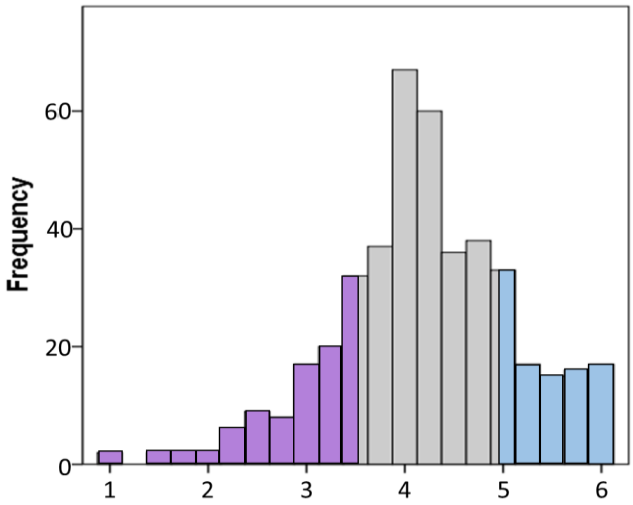


Figure 3.1. Distribution of bipolar scores for general beliefs in ability. In purple are individuals categorized as holding more entity beliefs (*M* minus one *SD)*; in blue are individuals categorized as holding more incremental beliefs (*M* plus one *SD).*

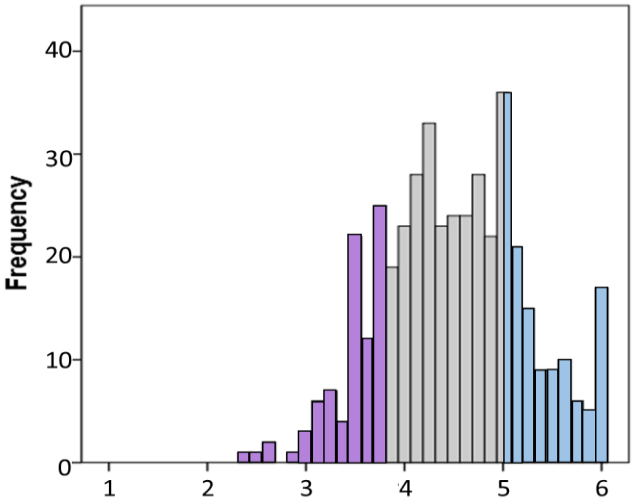


Figure 3.2. Distribution of bipolar scores for programming beliefs in ability. In purple are individuals categorized as holding more entity beliefs (*M* minus one *SD)*; in blue are individuals categorized as holding more incremental beliefs (*M* plus one *SD).*

3.2.2 General (and programming) mindsets: unipolar or bipolar constructs?

Traditionally, implicit theories (i.e., mindsets) have been assessed using up to four items that measure entity or incremental beliefs using a unipolar scale that prompts the respondent to think in terms of the presence or absence of the target construct. However,

differences in how scholars subsequently handle these items exist. Some researchers treat these two constructs as separate unipolar constructs for their subsequent analysis. For instance Erdley et al. (1997) and Hong et al. (1999) obtained a mean score for the entity items (and only used these items). Other researchers instead post hoc collapse the entity and incremental items into a single bipolar mindset construct, named the implicit theory scale, by reverse scoring one of the sub-constructs (e.g., entity) and obtaining an overall mean across all items (Blackwell et al., 2007). Recently, researchers have begun to question whether it is appropriate to collapse the two types of mindset into a single construct (Tempelaar et al., 2015). In particular, bipolar scales should have close to a perfect correlation of 1 between the two poles (McCroskey & Richmond, 1989, p. 160). Thus, reducing the entity and incremental constructs into a single continuum should only be done if there exists a strong correlation between entity and incremental constructs, something echoed by recent work (Tempelaar et al., 2015; Schweizer & Schreiner, 2010).

To shed more light on this issue and guide our subsequent analysis for phase 2, we conducted a correlation between each pair of mindset variables. Consistent with prior findings, (Tempelaar et al., 2015; Blackwell et al., 2007) we found negative correlations between incremental and entity beliefs for both general and programming mindsets, indicating that participants who endorse entity beliefs do not embrace incremental ones and vice versa (general entity and incremental: $r(435) = -.66$, $p = .01$; programming entity and incremental: $r(435) = -.46$, $p = .01$). However, of interest here, the correlations were only moderate, suggesting against post hoc collapsing the two constructs into a single scale by reverse scoring one of the constructs, as is done in some research (Blackwell et al.,

2007). Thus, we decided to keep the two beliefs as separate unipolar constructs for the analysis presented in Chapter 4, rather than collapsing them into a single mindset construct.

3.2.3 Exploring the relationships among constructs and beliefs

As the first step for evaluating the relationships between the various constructs, we ran a series of correlations between each pair of constructs (12 constructs for a maximum of 66 associations between constructs). To control for Type I error in the presence of multiple comparisons, we adjusted the alpha level using Bonferroni (adjusted $\alpha = .000758$). Table 3.2 presents the results. Here, we will focus only on the key findings.

The strongest correlation was between entity programming mindset and entity effort beliefs, indicating that students who believe that programming ability is fixed also believe that investing effort is evidence of their lack of programming ability. We also found a strong correlation between entity general beliefs and entity programming beliefs indicating that those participants who have the idea that abilities in general are fixed, also have a similar belief for computer programming. This is consistent with previous research, where associations between general ability beliefs and domain specific mindsets were found, albeit not with perfect associations (Scott & Ghinea, 2014). The results for incremental construct mirror for the results for the entity construct, although the former displays slightly weaker associations, suggesting the presence of more noise in the items measuring incrementally-related variables. In contrast to these high associations between mindset and effort, the goal constructs presented weaker associations and fewer significant associations.

To explore the relationships between constructs in a single model, we used a linear regression, with the goal of descriptive modeling (rather than casual inferences or

prediction). Two linear regressions were conducted, one for entity and other for incremental programming beliefs as the outcome variable, with the corresponding entity or incremental constructs as the predictor variables using the enter method (i.e., predictors considered were entity or incremental general intelligence beliefs, learning / performance approach / avoidance goals, entity or incremental associated effort beliefs, helpless attributions and positive or negative strategies, depending on whether the outcome variable was entity or incremental).

Table 3.2

Correlations between all constructs assessed in the online survey.

Variable	1	2	3	4	5	6	7	8	9	10	11	12
1. Entity general beliefs	–	-.66*	.53*	-.27*	.54*	.15*	-.19*	.32*	.18*	.32*	-.15*	.37*
2. Incremental general beliefs		–	-.13*	.44*	-.17*	.35*	.40*	-.05	.05	-.12*	.18*	-.14*
3. Entity programming beliefs			–	-.46*	.74*	.09	-.11*	.34*	.19*	.34*	-.17*	.44*
4. Incremental programing beliefs				–	-.28*	.34*	.38*	-.05	-.05	-.13*	.25*	-.17*
5. Entity effort beliefs					–	-.21*	-.17*	.46*	.25*	.46*	-.27*	.54*
6. Incremental effort beliefs							.52*	-.05	-.01	-.16*	.31*	-.21*
7. Mastery goals							–	-.03	-.15*	-.16*	.22*	-.26*
8. Performance avoidance goal								–	.33*	.31*	-.16*	.31*
9. Performance approach goal									–	.19*	.10*	.17*
10. Helpless attributions										–	-.26*	.55*
11. Positive strategies											–	-.43*
12. Negative strategies												–

Note. All data was Bonferroni adjusted. * represents significance at p = .000758 (two tails)

For the entity-related analysis, the overall model was significant ($F(10, 425) = 97.55$, $p < .001$, $R^2 = .53$, $R^2_{Adjusted} = .58$). Two variables were significant predictors (see Table 3.3), including entity general belief, and entity effort belief. Of these two, entity effort belief had the strongest association with entity programming belief.

Table 3.3

Beta coefficients from the regression model with entity programming beliefs as an outcome variable.

Variable	Standardized Beta Coefficients	Sig.
Entity effort	.64	< .001
Entity general beliefs	.18	< .001
Negative strategies	.05	.209
Helpless attributions	- .03	.408
Performance avoidance goal	- .02	.536
Performance approach goal	< - .01	.948

For the incremental-related analysis, the model's overall fit was somewhat reduced, in contrast with the one for entity beliefs, but still significant ($F(10, 425) = 40.03$, $p < .001$, $R^2 = .27$, $R^2_{Adjusted} = .26$); all variables were significant predictors (see Table 3.4), including incremental general and entity beliefs, learning goals and positive strategies; of these four predictors, incremental general beliefs had the strongest association with incremental programming belief.

Table 3.4

Beta coefficients from the regression model with incremental programming beliefs as an outcome variable.

Variable	Standardized Beta Coefficients	Sig.
Incremental general beliefs	.32	< .001
Mastery goals	.17	.001
Positive strategies	.12	.005
Incremental effort	.11	.035

3.2.4 Overall summary

In sum, we found little evidence of the 40/20/40 mindset distribution advocated by Dweck, indicating that there are less entity oriented individuals than predicted by prior research. We also did not find strong evidence that mindset should be treated as bipolar by reverse scoring one of the scales, and so in the analysis of data from the lab study in the next chapter, we treat the two mindset constructs as separate. Finally, our exploratory regression analysis indicated that entity programming beliefs are most strongly associated with entity effort beliefs, but also related to general entity belief in ability (with comparable findings for incremental programming belief).

Chapter Four

Phase Two: Laboratory Session

The goal of the second phase of the research was to evaluate the impact of an incremental mindset intervention in the context of an authentic instructional activity. Our research questions were as follows:

(1) Does an incremental mindset intervention change the beliefs of entity-oriented participants regarding their programming abilities? Prior research in implicit theories found that general beliefs can be changed in academic domains, however little work exists for the programming domain.

(2) Does an incremental mindset intervention impact participants' behavior? In contrast to prior work that has relied on self-reported variables like effort (e.g., Hong et al., 1999), we will address this question by measuring the time participants actually dedicate to programming activities, using it as a metric for persistence.

(3) Does an incremental mindset intervention impact performance? While there is work looking at academic performance in studies carried out in classrooms over the school year (e.g., Blackwell et al., 2007; Paunesku et al., 2015), little work exists using authentic instructional activates in laboratory studies.

To address these questions, an incremental mindset intervention was implemented under controlled laboratory conditions – we now describe the methods used in our study.

4.1 Method

4.1.1 Participants and recruitment

Participants were recruited via the online survey (phase one). Recruitment was more challenging than anticipated as participants had to be at least somewhat entity oriented in their programming beliefs, self-report low or no familiarity with programming concepts, and be willing to attend a two hour laboratory session. Out of the 143 students in the online survey who qualified to participate in the laboratory session, only 49 signed up for and attended the laboratory session. Two participants were excluded from the analysis (one decided in the middle of the session not to complete it, and a second did not follow instructions and was continuously asking questions about how much more work was still needed). Therefore, the final sample size was 47 participants (31 female). As a token of appreciation, participants received either $20 (51.10%) or 2% bonus credits towards a course (CGSC 1001; 48.90%). As mentioned in Section 3.1.1 we received research ethics board clearance.

All participants were over 18 years of age (M = 20.40; SD = 4.10) and the sample was varied in ethnicity: 46.80% were Caucasian/European, 14.90% Black (e.g., African American), 10.60% Arab (e.g., Saudi), 10.60% South Asian (e.g., Indian) and the rest were distributed among East Asian (e.g., Chinese), Southeast and West Asian (e.g., Filipino and Afghan, respectively). Around half of the sample (53.20%) was freshmen and the rest were approximately evenly distributed across the second, third and fourth year of studies. With regards to their programs and majors, the sample was also varied: 21.30% of participants were in psychology and the remainder were spread among 21 different programs and

specializations (e.g., criminology, communications, health science, child studies). Data collection lasted a bit more than two months (from February 7th, 2017 to April 13, 2017).

4.1.2 Materials

The following materials were used: (1) two brief research style articles, (2) two writing activities related to the reading materials, (3) the Student Questionnaire (described in Chapter 3, Section 3.1.2), (4) programming tutorials, and (4) two programming exercises. In subsequent text, when we refer to "programming activities", they include "programming practice" (from the tutorials) and the "programming exercises" (from the last part of the session). Overall, material preparation took around three months.

Research-style articles. Based on previous work (Blackwell et al., 2007) two articles were created for the present study, one for each condition (experimental and control). Both articles were designed to look like research articles; the length of each was four pages, designed to take no more than 10 minutes reading time. Once they were created, there were three rounds of pilots to make sure the key ideas were clearly communicated and the total reading time was similar / fit within the time frame planned. The contents of each article followed prior work from Blackwell et al. (2007); the control article, titled "*Habits you can Use to Study More Efficiently",* discussed study habits and strategies and how these can help students to become more effective. Also in line with previous work from Yeager et al. (2016), Paunesku et al. (2015), and Blackwell et al. (2007) the experimental article, titled "*You can Increase Your Intelligence*", provided scientific concepts related to changing core beliefs about the malleability of intelligence. The content of the article was tailored to the programming domain, following the advice from Yeager and Walton (2011), who highlighted the importance of customizing mindset interventions to the target domain. We

took care to develop both articles to be as similar as possible. Both have similar formats: a title, abstract, references, the same length, a two column format; both required similar amount of time to be read. We recognize that some differences might exist (e.g., the number of sub-headings), but we do not anticipate these to bias any results. Both articles are in Appendix E.

Writing activities. Two writing activities were developed based on prior work from Yeager and Walton (2011), Yeager et al. (2016), and Aronson et al. (2002) – the activities were the same for the two conditions. The first activity asked participants to write about what they learned from the article they read ("Please describe below what you got from reading the article – include the key ideas"). This was done to make participants review their understanding and to increase the chances that they processed the contents of the article. In order for participants to adopt the message in the article, it is also key to promote support for the content of the message, by putting into practice the "saying-is-believing-effect" (Aronson et al., 1999, 2002; Yeager & Walton, 2011). Consequently, the second writing activity requested the participants to write a letter to another student who was struggling with his/her studies to encourage that student to continue, even in the face of difficulty or challenge ("Please write a letter for another student who is experiencing difficulties in their courses. Specifically, what would you tell the struggling student to encourage him/her to keep going and not to drop out?"). Furthermore, the instruction for this second writing activity encouraged participants to use examples from the research article and or from their own experience ("Feel free to use the information that was provided in the article and/or to give examples from your own experience") (see Figure 4.1). Doing so is important because it helps to maximize the belief perseverance among

participants (Aronson et al., 2002). Both writing activities requested participants to write in full sentences avoiding bullet point formats.

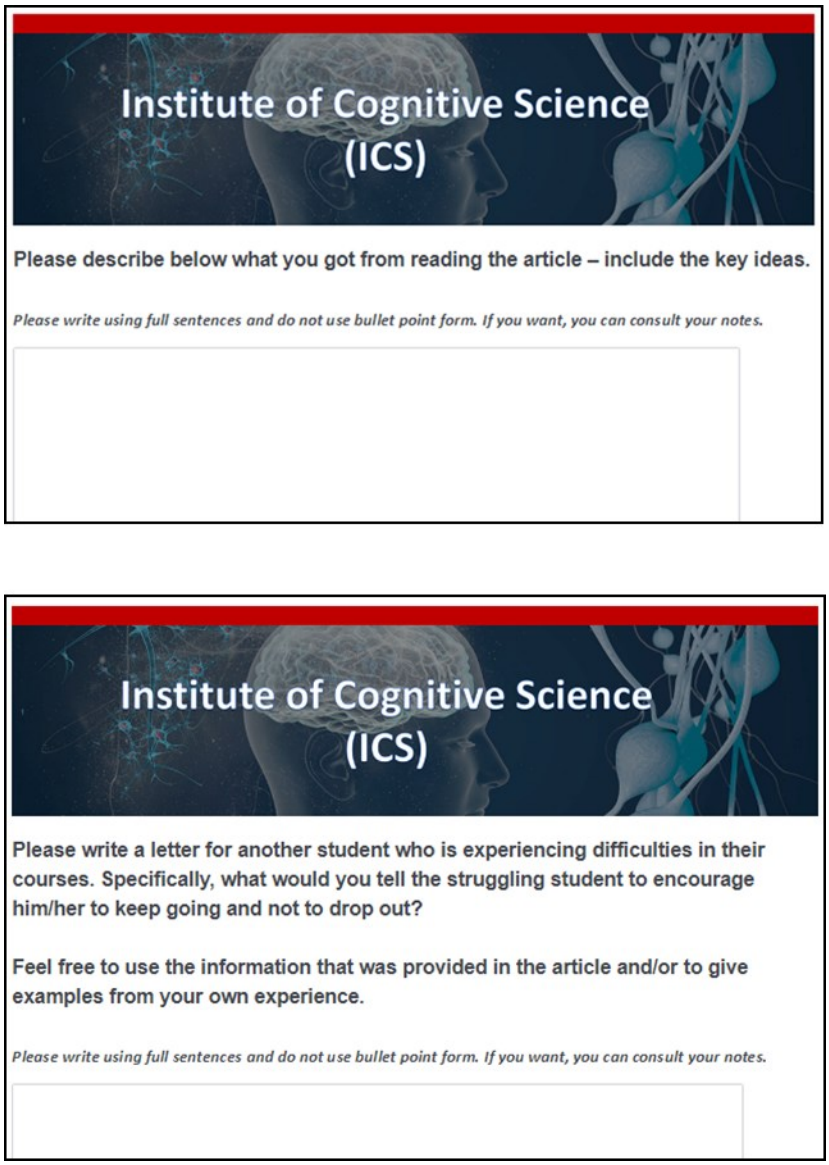


Figure 4.1. Writing activities (activity 1, top; activity 2, bottom).

Instruments to measure mindset and related factors. We used the same "Student Questionnaire" to measure mindset and related factors as for the first phase of this research (see Chapter 3, Section 3.1.2), including questions to measure (1) general implicit theories

of intelligence and ability, (2) customized implicit theories of ability for the programming domain, (3) effort beliefs, (4) achievement goals, and (5) helpless attributions and strategies in response to failure.

Programming tutorials. The tutorials were presented in an html file that contained both text and instructional videos. We designed the tutorials to guide participants in the acquisition of four standard programming concepts that progressed in difficulty, including: (1) variables – creation and display, (2) data input and display, (3) loops, and (4) conditionals. To ground these concepts in an authentic programming activity, the instruction for these concepts was centered on coding a guessing game in a visual, drag and drop programming language called Snap! This programming language was created by the University of California at Berkley to help students to learn to code (see Figure 4.2). Programming in Snap! is done by "snapping" blocks of code, allowing users to focus on code semantics rather than syntax; Snap! also allows its users to create a wide variety of control structures such as loops and conditionals. The html file with the tutorials was divided into four sections, one for each topic (see Figure 4.3). A brief introductory text was presented before each video. The first topic had two videos to allow participants to warm up in the process of the new programming concepts - the remaining three topics had one video. Each video showed a screen capture of a Snap! program being created (e.g., blocks being dragged over and snapped into place), with a narration describing the underlying concepts. The videos encouraged participants to actually perform the programming actions in Snap! (by following along during the video, as well as after) -this was to improve the acquisition of the concepts given that deliberate practice improves performance (Campitelli & Gobet, 2011). The total length of the videos was 17 minutes and 20 seconds. To make

sure the programming tutorials were clear, understandable, and conveyed the key concepts, several rounds of pilot tests were conducted; these are described in the procedure section.

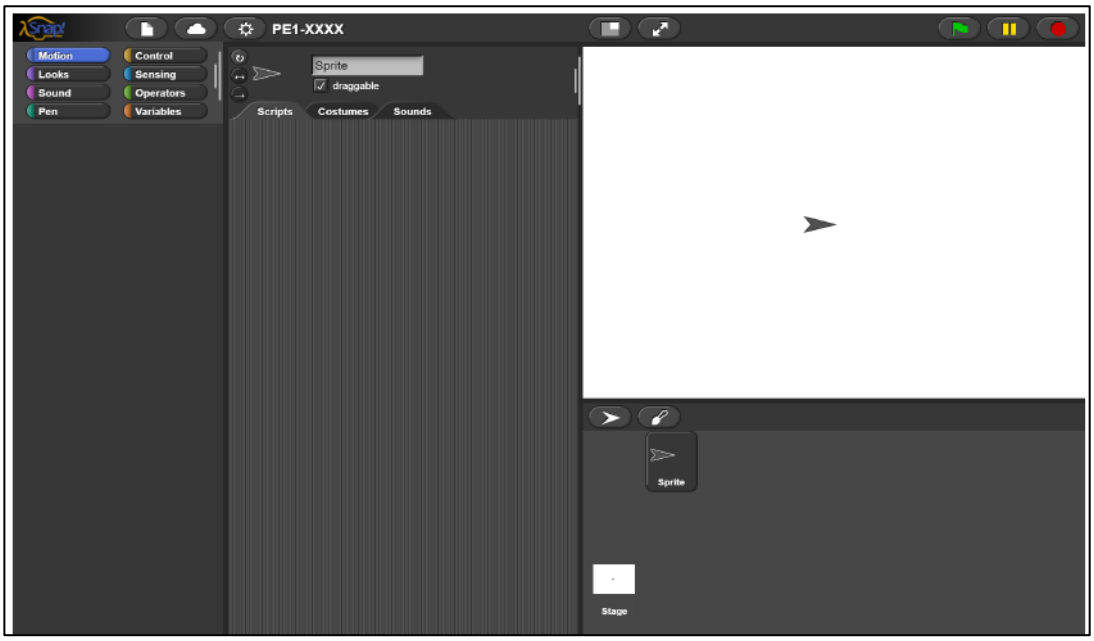


Figure 4.2. Snap! environment showing a standardized neutral template – this was used by participants to build their programs.

Programming exercises. To assess participants' acquisition of programming concepts two types of programming exercises were created, namely *similar* and *transfer* (these are in Appendix F). The similar exercise asked participants to recreate the final guessing game program from the tutorials, but from memory and without having any access to the tutorial material. The transfer exercise asked participants to make three modifications to provided code, such as adjustments to keep incrementing the value of a variable inside a loop ("Change the program so that it keeps incrementing the variable "age" forever") or making a program stop when the variable reached certain value ("Modify your program, so that it stops incrementing "age" once it becomes 100"). Instructions for the similar and transfer exercises were provided in the form of pdf files, with one file for each exercise.

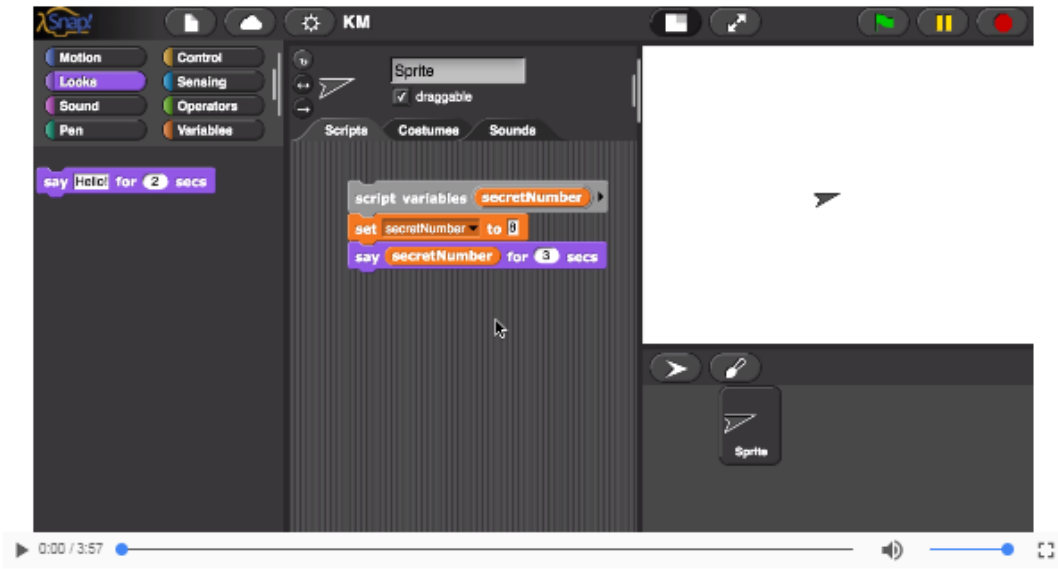
Topic Two: Getting information from the user

General Explanation:

Programs often need to get information from the user – like, for example, asking the user a question and storing their answer.

How to do it in Snap:

Please watch the video below to see how to get information from the user – for our program, this information will be about the user's guess for the secret number.



We encourage you to practice, by creating on your own, (not copying) the program in Snap that was shown in the video. Remember that once you finish reviewing about all the topics, you will be asked to write a program by yourself.

Topic Three: Loops

General Explanation:

A loop is a programming structure that repeats one or more instructions until a specific condition is met. For instance, we might want to keep asking the user to input some information until they have provided all the required information. The key aspect to programming a loop is figuring out which instructions should be repeated and thus placed inside the loop.

How to do it in Snap:

Please have a look to this video to see how we can program a loop needed for our secret number program.

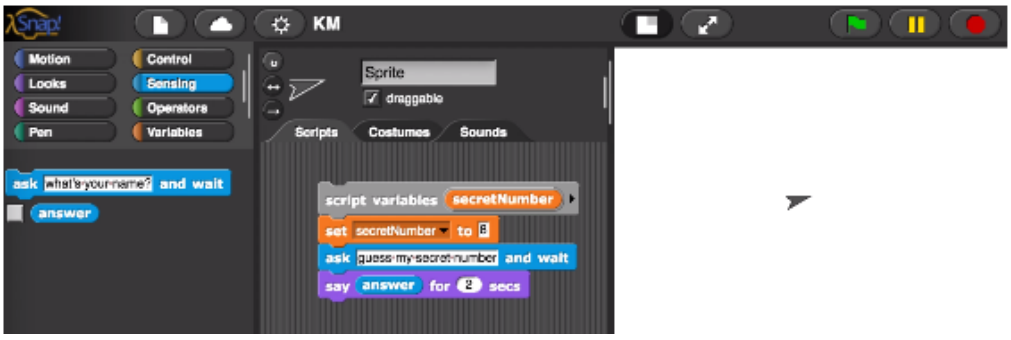


Figure 4.3. Fragment of the tutorial file containing explanatory videos and text.

4.1.3 Design

The study used a between subjects design with two conditions, control and experimental. Participants in both conditions performed the same tasks, in the same sequence. The only difference was the article that participants were asked to read ("*Habits you can Use to Study More Efficiently"* for the control condition and "*You can Increase Your Intelligence*" for the experimental condition). Thus, only the experimental condition received the incremental mindset intervention.

4.1.4 Procedure

During the study, the overall sequence of events was: (1) welcome and general instructions (3 min); (2) research article, related tasks, and "The Student Questionnaire" (30 min); (3) first break (3 min), (4) acquisition and practice of programming concepts via the programming tutorials (50 min), (5) second break (3 min), (6) programming exercises (similar and transfer) (30 min), and (7) debriefing (1 min) (see Figure 4.4). Except for the reading of the article, which was presented on paper, participants worked on a computer to perform all tasks. During the whole session, the researcher remained in the laboratory room with the participant, but sat with her back to them and pretended to be doing some other work. For all researcher/participant interactions an *a priori* script was used to ensure consistency between sessions. The length of the laboratory session was at most two hours. We briefly describe each step below.

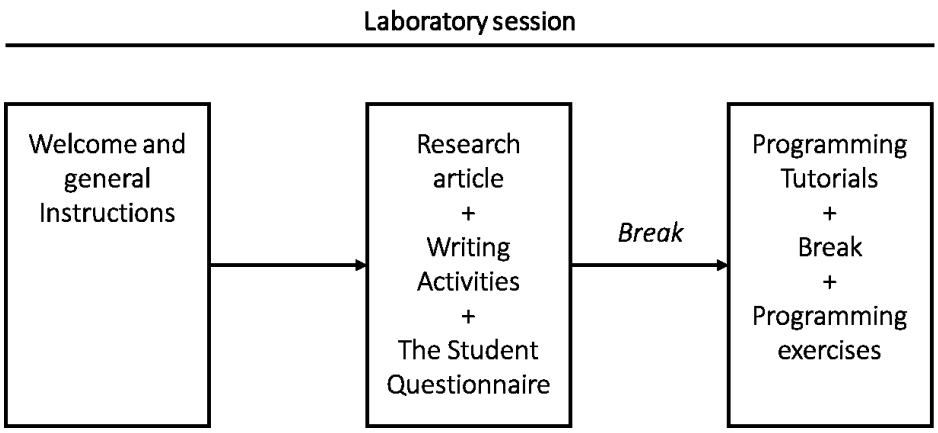


Figure 4.4. Overall sequence of events during the laboratory session.

Welcome and general instructions. Once participants arrived they were asked to read and sign the consent form. Then, the researcher provided some general instructions (e.g., putting in mute mode their mobile phones). The researcher also provided information about the length and flow of the session as well as general instructions about the use of the computer, such as not to open other applications or windows beyond the ones that were provided. Any questions that participants had were answered.

Research article, related tasks and "The Student Questionnaire". Participants were assigned to one of two conditions prior to arriving to the session. We used a matched assignment technique to control for gender and the recruitment source (SONA vs. external sources: posters and classroom announcements). During the study, participants were asked to read the respective research article ("*You can Increase Your Intelligence*" for the experimental, and "*Habits you can Use to Study More Efficiently"* for the control). In both conditions, to encourage participants to engage with the article, participants were permitted to make notes (directly on the article or on an extra sheet of paper) and to highlight the article. Based on prior work, reading time was controlled to be exactly ten minutes (Yeager et al., 2016). This was done to make sure participants went over the information in the

article with enough detail, minimizing the risk of missing important information; if participants finished reading the article before the time was up they were asked to review the article and their notes.

After reading the article, participants were asked to engage in the two writing activities related to the article. The instructions for these activities were read aloud by participants, to ensure that they were processed by participants. If clarification-type questions came up, the researcher answered them (e.g., the format of the letter participants were asked to write). The time allocated for these two writing activities was of a maximum of 10 minutes for each (participants could spend less time if they chose). This time was selected based on three rounds of pilot tests, as well as prior work implementing incremental interventions using these writing activities (Yeager et al., 2016; Aronson et al., 2002). None of the participants asked for more time to complete the two writing activities. Once participants finished the second writing activity they filled in the Student Questionnaire.

Programming tutorials. During the first break, the researcher opened an electronic document in Adobe with participant instructions. Participants were asked to read the instructions aloud to make sure they processed the information (see Appendix G); when participants finished reading the document, the researcher closed the file. Then, the researcher opened the programming activities: on the left side of the screen, Snap! with a standardized neutral template, used by all participants to do their programming practice; on the right side of the screen the html file with programming tutorials (see Figure 4.5). Participants were told that they could read the texts in the html file and to rewind or fast forward the videos as many times as they wished. Participants were also told that for the

programming exercises that followed the tutorials, they were not going to be allowed to use any aids (neither the videos nor their programs in Snap! they wrote while following along with the tutorials). Participants were given a maximum of 50 minutes for this period - this time included the videos, which were just over 17 minutes in length. All sessions were screen recorded using the Camtasia screen recording software.

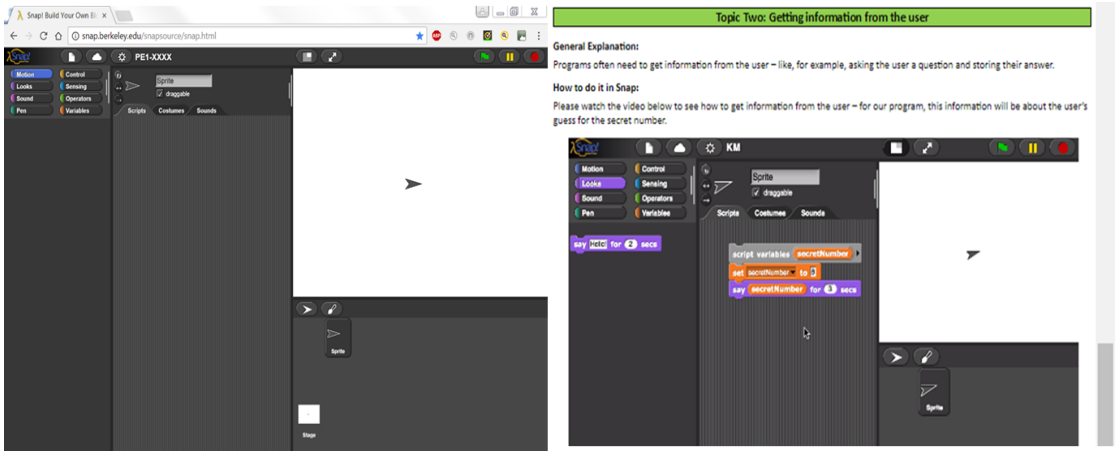


Figure 4.5. Screen look and feel during the tutorials: Snap! environment with a neutral template on the left, and the tutorials (html file) on the right.

Prior to the study, eight rounds of pilots were conducted to test the content and the flow of the tutorials and associated videos before they were recorded and four more pilots were conducted after integrating all the materials – these were used to test the materials only and not the full experimental set up and script. Accordingly, three final pilots were conducted at the laboratory, using the real environment and hardware. Several adjustments were implemented, such as text phrasing of the script and instructions to shorten or clarify, adjusting the tutorials to make them of appropriate length and/or complexity.

Programming exercises (similar, transfer). During the break that preceded the final stage of the study, the researcher opened a new Snap! generic template, which participants

used for the *similar* exercise, and the Adobe file with the corresponding instructions, each occupying half of computer's screen (see Figure 4.6). Participants were reminded not to open new windows or files, and to only work in the windows and applications provided. After participants finished with the *similar* exercise, the researcher saved and closed the Snap! file and opened a new Snap! neutral template, as well as the associated Adobe file with instructions for the *transfer* exercise (see Figure 4.7)

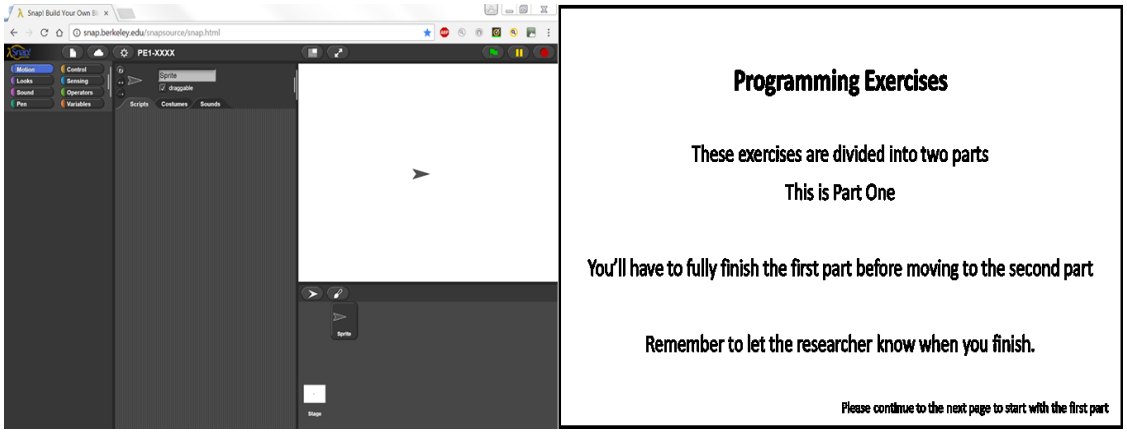


Figure 4.6. Screen look and feel during the similar exercise: Snap! environment with a neutral template on the left, and instructions for the similar exercise (Adobe file) on the right.

During this stage, no other instructional aid was permitted, especially because similar exercise asked participants to rewrite, on their own, the code for the guessing game program. However, participants had some resources. For instance, Snap! provided all the necessary code building blocks. For the similar exercise, participants were given a high level image of how the solution should look like (without actually showing the programming code). Also, for the transfer exercise, participants were provided with two aids to support their work (see Appendix F). Participants were given a total of 30 minutes

for the *similar* and *transfer* exercises; if they finished their two exercises before time was up, they were instructed to let the researcher know.

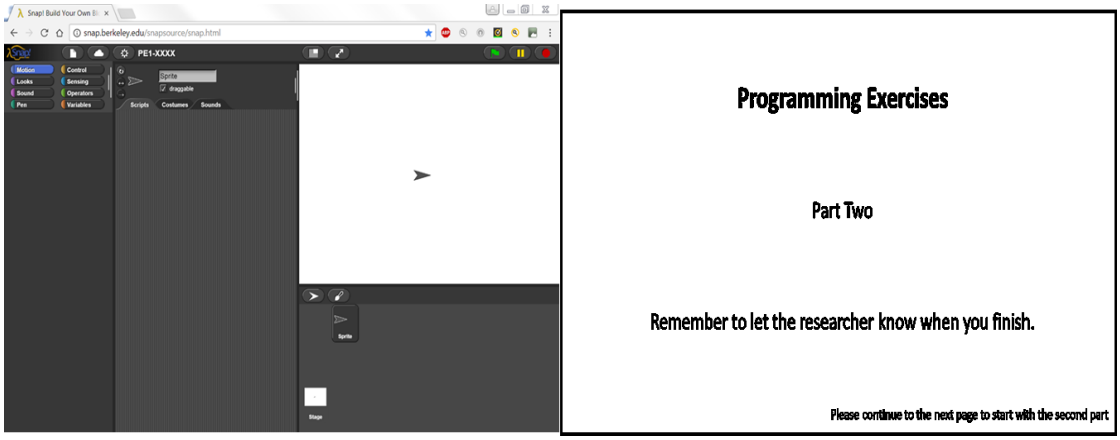


Figure 4.7. Screen look and feel during the transfer exercise: Snap! environment with a neutral template on the left, and instructions for the transfer exercise (Adobe file) on the right.

4.1.5 Dependent Measures

To investigate the impact of the mindset intervention we used the following dependent measures: (1) changes in implicit theories, (2) persistence, and (3) performance on the two programming exercises. Variable operationalization for points two and three were made based on behavioural variables rather than self-reports. Below we describe each one.

Change in implicit theory. Change in implicit theory was operationalized as the difference in participants' mindset beliefs measured prior to the experiment via the online survey and the measure obtained in the laboratory session after the writing activities.

Impact on persistence. We measured persistence through the time participants dedicate to their activities. Therefore, we operationalized persistence in terms of the time

dedicated to the programming activities (including the tutorials and the two exercises). Time was assessed by using the screen recordings.

Impact on programming performance. To obtain a total score for each of the two exercises (similar and transfer) we analyzed participants' final solution to each exercise, based on the following two aspects: (1) presence of the correct individual programming instructions: *individual commands* component (accounting for the 40% of total score) and (2) *holistic* component assessing overall program functionality (accounting for the 60% of the total score). More weight was given to program functionality (60%) rather than to the correct instruction selection (40%) because the end goal of the exercises was for participants to build their own functional programs, either based on memorizing of the use of the instructions (in the similar exercise) or by generalizing the instructions to other contexts (in the transfer exercise). To ensure consistency, a detailed coding scheme was constructed for each aspect (individual, holistic) - see Appendix H. Whereas for the *individual commands* component, the coding scheme was based on whether the correct program instructions were merely present, for the *holistic* component, the scheme considered whether the instructions were placed in the correct position and in a way that the program would run as it was expected to. To maximize grading objectivity, we blinded all materials before grading them by using the identification numbers of four digit codes (without any indication of the condition); we also made multiple passes to ensure consistency. For the grading, we implemented a principled grading process, where two key activities were performed: (1) each solution was isolated by visiting the screen capture video and printed, (2) the videos and the Snap! files were consulted for each participant, to ensure consistency.

4.2 Results

We now present the results related to our three research questions about the mindset intervention's impact on beliefs and performance.

4.2.1 Change in implicit theories

To examine the impact of the mindset intervention on implicit theories, the analysis will be centered on the programming beliefs (entity and incremental), as they are the focus of the present research. However, for exploratory purposes we also ran analysis for the effort and general ability beliefs. Table 4.1 shows the means and standard deviations for these variables before and after the incremental intervention and control activity (for the sake of completeness, the descriptives for the other measured constructs are in the Appendix I. First, we verified that no differences existed between the two conditions prior to the intervention (as reported by an independent samples t-test – no significant differences were found, see Table 4.2). Notably, for each of the target constructs, the absolute mean differences (computed by subtracting the post interview survey mean from the pre interview survey mean) resulted in larger absolute differences in the experimental condition than in the control condition, indicating that the mindset intervention changed beliefs in the desired direction more in the experimental condition.

Table 4.1

Means, standard deviations and mean differences for the programming beliefs before and after the incremental mindset intervention and control activity (top) and for the general and effort beliefs (bottom).

Variable	Experimental (*N* = 24)					Control (*N* =23)				
	Pre		Post		*M* Post - Pre	Pre		Post		*M* Post - Pre
	M	*SD*	*M*	*SD*		*M*	*SD*	*M*	*SD*	
Entity programming beliefs	3.70	.30	4.48	.85	.78	3.84	.47	4.13	.78	.29
Incremental programming beliefs	2.73	.69	2.14	.86	-.59	2.83	.72	2.74	.86	-.09
Entity general beliefs	3.80	.98	4.61	.98	.81	3.85	.89	4.04	1.08	.20
Incremental general beliefs	2.88	.72	2.14	.90	-.74	2.92	.94	2.79	1.06	-.13
Entity effort	3.77	.57	4.40	.79	.63	3.92	.67	4.18	.91	.26
Incremental effort	2.36	.80	2.04	.66	-.32	2.74	.74	2.60	.84	-.14

Table 4.2

T-tests between experimental and control conditions for target constructs prior to the reading and writing activity.

Variable	t-test	
	t	Sig. (2-tailed)
Entity programming beliefs	1.20	.236
Incremental programming beliefs	.47	.638
Entity general beliefs	.17	.868
Incremental general beliefs	.20	.841
Entity effort	.85	.398
Incremental effort	1.66	.104

df = 45

To verify the impact of the intervention on programming beliefs, two two-way mixed ANOVAs were conducted with the programming beliefs as the dependent variable (one ANOVA for entity and one for incremental belief), *condition* (experimental vs. control) as the between-subjects independent variable, and *point in time* (pre and post intervention) as the within-subjects independent variable. Of primary interest here is the interaction between *condition* and *point in time*, and this was significant in both analyses (see top part of Table 4.3). Participants in the experimental condition experienced a larger change in their mindset than those in the control condition, becoming less entity oriented and more incremental oriented, respectively (see Figure 4.8).

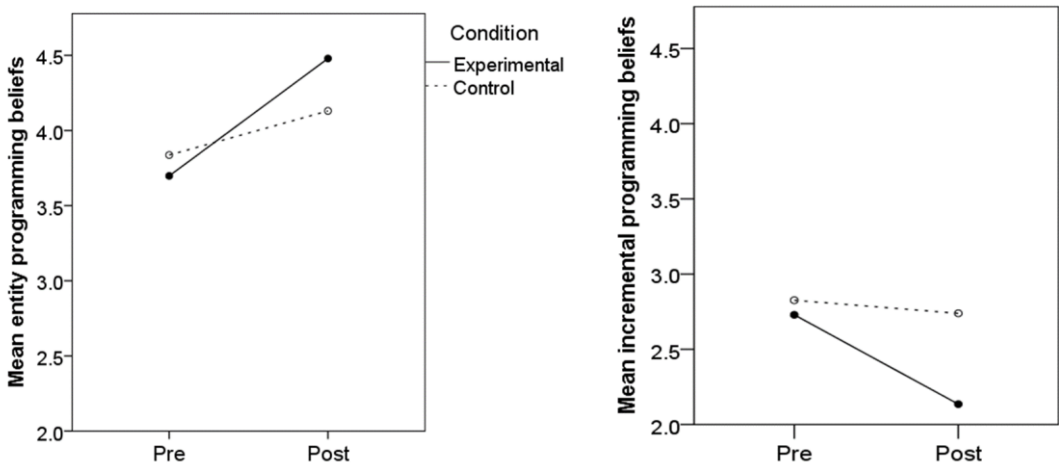


Figure 4.8. Interaction between time and condition for change in entity and incremental programming beliefs

For exploratory purposes, to see whether the intervention had an effect on participants' general mindset and effort beliefs, we conducted four more two-way mixed ANOVAs with these beliefs as the dependent variable, i.e., entity general mindset / general mindset incremental / entity effort beliefs / incremental effort belief (in each ANOVA, *condition* was the between-subjects independent variable, and *point in time* was the within-subjects independent variable). For general mindset beliefs, the results were similar to those for the programming beliefs, with participants in the experimental condition experiencing a significantly larger shift in mindset (e.g., reducing their belief in entity theory, see bottom part of Table 4.3). We found a trend that participants reduced their belief in entity-effort belief more in the experimental condition than control (the intervention did not significantly impact incremental effort belief). The rest of the interaction graphs are in Appendix K.

Table 4.3

Interaction between time and condition for change in implicit theory (programming beliefs, top; general and effort beliefs, bottom)

Variable	*F*	Sig.	Eta squared
Entity programming beliefs	5.1	.029**	.07
Incremental programming beliefs	5.0	.030**	.09
Entity general beliefs	7.2	.010**	.10
Incremental general beliefs	8.9	.004**	.12
Entity effort	3.0	.091*	.05
Incremental effort	.8	.392	.02

Note. ** Significant at $p < .05$. * Trend at $p < .1$. For all variables $df = 1, 45$

4.2.2 Impact on persistence

To examine the impact of the mindset intervention on persistence, we extracted the time participants dedicated to the programming activities (tutorials, similar, and transfer exercises). As shown in Table 4.4, overall mean time in the experimental condition was higher than in the control condition. To assess the effect of the incremental mindset intervention on the time spent in the programming activities, we ran a two-way mixed ANOVA, with the time dedicated as the dependent variable, *condition* (experimental vs. control) as the between-subjects independent variable, and the programming activity (tutorial, similar exercise and transfer exercise) as the within-subjects independent variable. There was a marginally significant main effect of condition on the total time that participants dedicated to the programming activities (F (1, 45) = 3.87, p = .055, η^2 = .08) indicating that participants spent more time in the experimental condition than in the control condition. We did not find evidence that condition interacted with programming activity to impact the amount of time spent as indicated by the lack of significant interaction

between condition and programming activities on time spent on task F (2, 45) = .11, p = .893, $\eta^2 < .01$.

Programming Activity	Experimental		Control	
	M	*SD*	*M*	*SD*
Practice	1642.54	460.11	1436.91	345.39
Similar exercise	469.67	423.46	356.43	224.47
Transfer exercise	976.54	451.27	860.13	363.96
Total	3088.75	845.27	2680.48	537.16

Thus, overall our results provide some evidence that the incremental manipulation increased persistence devoted to the programming activities. We also checked time devoted to each activity, with a focus on the transfer activity. We designed the transfer activity to be more challenging and this was verified by our analysis - there was a significant main effect of activity on time (F (2, 45) = 108.40, $p < .001$; $\eta^2 = .71$), and participants spent longer on the transfer exercise than the similar exercise ($p < .001$) suggesting that participants found the transfer exercise more difficult. However, we recognize that time alone is not sufficient to determine difficulty and our subsequent analysis on performance provides further evidence for the difficulty of the transfer exercise.

4.2.3 Impact on programming performance

To evaluate the impact of the incremental mindset intervention on participants' programming performance, we scored the two exercises using the method described above. Table 4.5 presents the score means and standard deviations as a function of condition. As expected the mean scores were higher for the similar exercise than the transfer exercise. This was confirmed by a two-way mixed ANOVA, with *score* as the dependent variable,

condition (experimental vs. control) as the between-subjects independent variable, and the type of *programming exercise* (similar and transfer) as the within-subjects independent variable. Specifically, there was a significant main effect of exercise on score (F (1, 45) = 12.31, p = .001, η^2 = .20) indicating that collapsed across condition higher scores were obtained for the similar exercise than for the transfer exercise and suggesting that transfer exercise was indeed harder than the similar one.

Table 4.5

Means and standard deviations for the scores obtained (max score = 10) in the programming exercises as a function of condition

Programming Exercise Score	Experimental		Control	
	M	*SD*	*M*	*SD*
Similar	8.29	3.36	8.90	1.80
Transfer	6.68	3.20	7.79	2.11

Of primary interest was whether condition influenced how participants' performed on the programming activities. We did not find evidence of an impact. The main effect of condition on score was not significant (F (1, 45) = 1.15, p = .289; η^2 = .03) and neither was the interaction between condition and score (F (1, 45) = .00, p = .989, η^2 < .01).

4.2.4 Influence of prior programming experience

During the random assignment of participants to conditions, we controlled for key variables (such as gender and recruitment source). We did not control for prior programming experience because we aimed to recruit participants with little or no experience, and because matching across multiple variables would have been impractical. However, by chance, more naïve programmers, ones with self-reported "none" as the

programming background, were assigned to the experimental condition (see Table 4.6). Therefore, to investigate whether this influenced our results we conducted an exploratory analysis, where we added *prior experience* (none, some) to a model that also contained *condition*. Our primary interest was to see whether condition and prior experience interacted with each other to impact the outcome variable in each of the three analyses performed: analysis 1 - change in belief, analysis 2 - impact on persistence, analysis 3 - impact on performance. Note that these analyses mirror the three done above (except that we now include prior experience in the model). To simplify the reporting of the results, we collapsed across multiple instances of the dependent measure for all three analyses, using the difference in entity programming beliefs for analysis 1, the total time dedicated in analysis 2, and the total score for analysis 3.

To assess the joint effect of prior experience and condition on change in belief

Table 4.6
Number of participants with some or no prior programming experience

Condition	No experience	Some experience
Experimental	18	6
Control	13	10

(analysis 1), we ran a between subjects ANOVA, with *difference in entity programming beliefs* (post minus pre) as the dependent variable and, as previously mentioned, *prior programming experience* and *condition* as the two factors. Focusing on the interaction, we did not find evidence that prior programming experience interacted with condition (F (1, 43) = 1.44, p = .237, η^2 = .03).

To assess the joint effect of the prior experience and condition on the total time spent in programming activities (analysis 2), we ran a second between subjects ANOVA. The

total time spent on programming activities was the dependent variable, and as above, *prior programming experience* and *condition* were the two factors. We found a marginally significant interaction between prior programming experience and condition on the total time on task (F (1, 43) = 3.17, p = .082, η^2 = .26). As shown in Figure 4.9, participants in the experimental condition who had no prior programming experience dedicated the most time to their activities overall and more than their peers in the same condition but with some programming experience. In contrast, participants in the control condition, regardless of their programming experience, dedicated similar amounts of time to their programming activities – and of key note, less than those with no programming experience in the experimental condition.

Finally, to assess the joint effect of prior experience and condition on total score (analysis 3), we ran a third between subjects ANOVA, with *total score* as the dependent variable, and *prior programming experience* and *condition* as the two factors. Here, the interaction between condition and prior programming experience on the total score participants obtained was not significant (F (1, 43) = .50, p = .484, η^2 = .01), suggesting that our prior result related to the lack of condition's impact on scores was not biased by prior programming experience.

In sum, this additional exploratory analysis showed that the slight imbalance that participants had in prior programming experience, assigned by chance between conditions, did not influence the change in beliefs (analysis 1) nor the scores (analysis 3). While the prior experience did have an effect on time spent, this was only the case for the experimental condition, which is an encouraging findings, since it suggests the intervention may have helped to increase persistence.

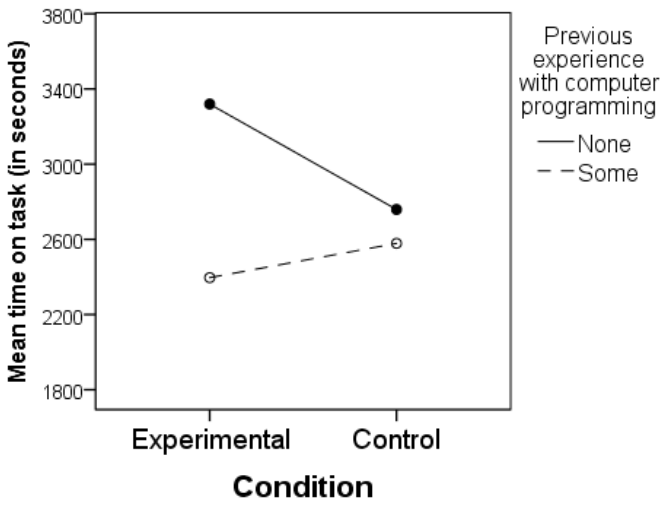


Figure 4.9. Interaction between prior programming experience and condition on total time dedicated to the programming activities

4.2.5 Overall summary

The key outcomes of the incremental mindset intervention were that it significantly reduced the endorsement of the entity programming (and general ability) beliefs, which in turn marginally increased persistence as measured by time on task. However, this did not translate to improvement in performance.

Chapter 5

Discussion

A key result of our work is that the incremental mindset intervention made participants become significantly less entity oriented in terms of their beliefs in ability for the programming domain (.6 to .8 absolute points), as compared to the control, and larger effect sizes[5] in comparison to prior work. Some prior work[6] has also successfully manipulated participants' beliefs, reducing their belief that ability is not malleable, with lasting effects from 2 weeks to up to eight months (Yeager et al., 2016). However, as will be highlighted by our review below, this other work has predominately employed multiple sessions and required more time. Much less work exists on the effectiveness of shorter mindset interventions and so our work contributes to the field by evaluating such an intervention.

Cutts et al. (2010) implemented a mindset intervention in the programming domain for students learning to program in python (presumably college participants, although the authors are not explicit about participants' demographics). During the first four weeks of the school semester, tutors dedicated the first 10 to 15 minutes of the two hour weekly lab sessions to the intervention. After seven weeks, students who received the intervention

[5] In our study, for the change from pre to post in each conditional separately, our Cohen's d was.*8* or bigger - prior work for that type of comparison reported $d = .66$. For the comparison between experimental and control using the differences from pre to post, our Cohen's d was larger than .45 for all significant findings; prior work for same comparison reported $d = .39$. See Appendix J.

[6] Using also 1 to 6 point Likert scales

became significantly more incremental in their beliefs[7] whereas those in the control condition became more entity-oriented over time. Cutts and his colleagues (2010) argue that this latter result might be due to ample opportunities for errors students have when learning to program. Therefore, "students may get stuck at many different stages and could interpret this to mean that they do not have the ability." (p. 4). These results are also consistent with other work that aimed to affect general mindset (rather than domain specific mindset). For instance, Blackwell et al. (2007) implemented a mindset intervention consisting of eight 25 minute workshops each among 7th graders. This intervention made students oriented more towards the incremental mindset (by approximately .6 absolute points; and d = .66 for the experimental condition). Another example comes from Yeager et al. (2014), who aimed to change participants' belief in theories of personality via a mindset intervention that like in our work was a short intervention (roughly 25 minutes). The mindset intervention altered students' implicit theories, leading to lower entity theories of personality (by about .5 absolute points; d = .39) compared to participants in the control group. Other researchers have also found a positive impact of the mindset intervention on students' incremental and entity beliefs, such as Aronson et al. (2002), Good et al. (2003) and Paunesku et al. (2015) (although these works do not explicitly report the values that mindset shifted by as a result of the intervention). In sum, the vast majority of prior work demonstrated that incremental interventions are effective for promoting a reduction of entity beliefs.

[7] Change was in about 0.4 absolute points – while the exact range of the Likert scale was not specified, the standard is to use scales that range from 1 - 6

In our work, the effect of the intervention on participants' beliefs in terms of making them less entity oriented was bigger than reported belief shifts in prior work (Blackwell et al., 2007; Yeager et al., 2016). Our results are also promising given that in contrast to the majority of other studies, our intervention was short and delivered in a single session. Importantly, previous work (Robins & Pals, 2002) has found that implicit theories are relatively stable over time, and that although they can be changed via interventions (Aronson et al., 2002; Blackwell et al., 2007) once changed, they do not spring back (Yeager et al., 2014).

While it is encouraging that our intervention influenced participants' beliefs in their own programming ability, it is also important to look for associations between belief and behaviors, both self-reported and actual. On the self-report side, prior research (Blackwell et al., 2007) found that participants who endorse incremental beliefs also self-reported more positive beliefs about the value of effort. Consistent with these findings, but on the entity side, our online results found a strong correlation between entity programming beliefs and entity effort. In addition, prior work has also shown that participants exposed to incremental interventions have self-reported to be more likely to invest more effort in comparison to their control peers. As observed by Hong et al. (1999, Study 3) students self-reported to be more likely to take remedial courses (deploy effort) after being exposed to an incremental condition than their control peers (and after having received negative performance feedback). However, since self-reports are not always accurate, it is important to also examine its impact on actual behaviors, as we did. Our results related to actual behaviors showed that after being exposed to the incremental intervention, participants spent marginally longer on their programming activities than their control peers. Putting

all these findings together, our incremental intervention encouraged participants to adopt more positive beliefs in the value of effort, which may have motivated them to spend longer on the programming activities than their control peers. We verified that this result related to time on task was not driven by the presence of slightly more participants with no experience in the experimental than in the control condition. We found some indication that the intervention actually increased time on task the most for novice programmers, which is particularly promising.

While the mindset intervention may have encouraged persistence, this did not translate to performance: there was no difference between conditions in terms of the scores for the programming activities. This was not consistent with prior work by Blackwell et al. (2007) and Paunesku et al. (2015) – however, in those studies performance was measured as grades over the semester, and there was a positive correlation found between performance and mindset. In contrast to these studies but consistent with our work in the programming domain, Cutts et al. (2010) showed that over a period of six weeks, the growth mindset intervention changed students' mindsets but that this alone was not enough to improve test scores. Cutts argue that "one possible reason for this is that the computer science atmosphere is one where a growth mindset may be challenged repeatedly. Students come up against failure regularly, more than in most other subjects." (p.4). This argument has also been suggested by Murphy and Thomas, (2008) who mentioned that students learning to program "face many challenges and a barrage of negative feedback: unfamiliar tools and environments, cryptic syntax and run time error messages, and incorrect output caused by elusive logic errors." These circumstances may make entity mindsets "to likely interpret excessive negative feedback as a challenge to their intelligence and to avoid

similar situations in the future." (p. 272). As described by Perkins (1986), while some novice programmers take "the inevitable occurrence of bugs in stride", others "became frustrated every time they encounter a (program) problem." (p. 267). In addition, Cutts et al. (2010) findings showed that an improvement in scores was observed only when the mindset intervention was supplemented with incremental messages containing detailed feedback for each task assigned.

Grant & Dweck (2003) found that differences in performance between entity and incremental theorists are observed in the face of challenge and failure; in face of success, both entity and incremental mindsets keep their belief and achieve similar performance (Diener & Dweck 1978; Dweck 1999). Therefore, it is unlikely that situations of repeated success will modify peoples' beliefs. The programming tasks in our research were designed to be challenging and likely did involve some implicit setbacks, for instance when programs did not work as expected. However, the tasks did not put participants in an explicit failure situation, since they were not given explicit performance feedback. Thus, this absence of explicit negative performance feedback may provide a possible explanation for the lack of a significant effect in performance. Another possibility relates to time. Specifically, prior research found improvement in performance over a longer period of time than an experimental lab session. For instance, impact on performance was observed after one semester and up to two years, with academic performance *improving* (Paunesku et al., 2015), slowing down its *decrease* (Yeager et al., 2014) or even *reversing a decrease* (Blackwell et al., 2007). Thus, while immediate differences in implicit theories might be observed, impact on performance may require an incubation period to show improvements. In general, more research is needed to test this hypothesis.

Another possibility why performance was not impacted by the incremental mindset intervention might be because there may be a misalignment between what participants self-report (e.g., belief in value of effort and persistence) and what they exhibit as their actual behaviors. In our study, we did find some indications of an alignment between self-reported effort and behaviours, as participants dedicated more time to programming activities after an incremental intervention. However, this effort did not result in better performance, perhaps because participants did not possess the academic strategies and expertise needed to improve their performance or because they were not aware they failed. In sum, there are a number of reasons (such as the lack of explicit negative feedback, the absence of incremental messages reminders, or the potential need for "incubation time") that might explain why our intervention did not impact participants' performance.

We did not find that our results for the distribution of the general ability beliefs were consistent with previous reports. While this may be due to differences in the way these distributions were computed, we aimed to replicate as best as possible prior methods. Other potential explanations for these differences include differences in the population (e.g., Hong et al.'s (1999) participants were Chinese students), or differences in ages (Blackwell et al.'s (2007) participants were junior high school students). A final possibility relates to the natural evolution of these beliefs, where in today's society, people hold a more mixed approach than 10-20 years ago.

A final note

As previously mentioned, one way in which research in mindset interventions have been implemented is through school studies (Blackwell et al., 2007) - the other way has been via studies under controlled laboratory environments (Hong et al., 1999). Each has

advantages and disadvantages. School studies increase ecological validity but don't easily support experimental control or provide opportunities to measure fine grained details, such as time students dedicate to particular task, or how this metric is impacted by the incremental interventions. Lab studies address some of these limitations but to date have not relied on authentic educational activities and moreover, they typically have relied on self-reports rather than actual behaviors. In fact, to the best of our knowledge, there is no prior research that has measured the impact of an incremental mindset intervention on actual behaviors under laboratory conditions using authentic academic activities in the programming domain. One of our contributions is that we addressed this gap by having controlled laboratory conditions using a real academic activity (although we acknowledge that we lost some ecological validity through our design). Our study showed that (1) the incremental mindset intervention reduced entity beliefs, (2) the intervention had an effect (albeit marginal) on persistence. However, participants exposed to the incremental intervention did not exhibit a higher performance than their control peers. Thus, more targeted domain-specific support may be needed to improve performance and learning, over and beyond what the tutorials in the present work provided. In general, while mindset interventions certainly hold potential, they should complement rather than replace domain-based interventions.

5.1 Limitations and Future work

Prior work has reported improvements in performance in certain populations, such as students at risk (such African-American in Aronson et al., 2002) and underperforming students (Paunesku et al., 2015). We aimed to control for prior experience by screening participants to only include ones who self-reported having little or no previous

programming practice. However, we did not tightly control for previous participants' programming performance as we did not use a programming pre-test, which might have reduced our power to detect effects. Since our study was already fairly long, adding a pre-test requires moving to a multi session study, something that future work could do.

Another limitation is that our research did not test the long term impact of the intervention on the change of beliefs and performance (e.g., did students decrease in entity beliefs hold after several weeks or months following the study?). This would have been interesting to assess via a delayed mindset survey and a delayed programming activity evaluation. Prior work has shown that sometimes impact is only observed in delayed evaluations (Van Gog & Kester, 2012). However, this work was not in the mindset domain, and so it would be interesting to explore the effects of this technique in the context of an incremental intervention.

As far as modeling the relationships between variables (phase 1), our work was limited to fairly basic regression analyses. Prior work used more sophisticated structural equation modeling (Blackwell et al., 2007), which has the potential to shed further light on the data. Our research did not perform structural equation modeling, as we did not collect outcome variables for the online survey which was the largest data set we needed to run such analysis. Future work could address this constraint.

In terms of the measures we choose, another potential limitation relates to the way we operationalized persistence, because it does not take into account success. Thus, a compound metric could have been also used, by creating an index which considered not only the time participants spent to their programming activities (as we did), but also their failure or success. In general, we used a basic measure of persistence, i.e., time advocated

to the programming activities participants engaged in. It would be also interesting to do content analysis of the videos we screen recorded from participants' actual behaviors during their programming activities, to get more in-depth understanding of what was happening and to see where participants were devoting time (and why this did not translate to performance benefits).

References

Aiken, L. S., West, S. G., & Reno, R. R. (1991). *Multiple regression: Testing and interpreting interactions*. Newbury Park, Calif: Sage Publications.

Aronson, J., Lustina, M. J., Good, C., Keough, K., Steele, C. M., & Brown, J. (1999). When white men can't do math: Necessary and sufficient factors in stereotype threat. *Journal of Experimental Social Psychology*, 35(1), 29-46. doi:10.1006/jesp.1998.1371

Aronson, J., Fried, C. B., & Good, C. (2002). Reducing the effects of stereotype threat on African American college students by shaping theories of intelligence. *Journal of Experimental Social Psychology*, 38(2), 113-125. doi:10.1006/jesp.2001.1491

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, N.J: Prentice-Hall, Inc.

Berkley University, (N/A) Snap! Build your own blocks. Retrieved from: http://snap.berkeley.edu/

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

Burnette, J. (2010). Implicit theories of body weight: Entity beliefs can weigh you down. *Personality and Social Psychology Bulletin*, 36(3), 410-422. doi:10.1177/0146167209359768

Burnette, J. L., O'Boyle, E. H., VanEpps, E. M., Pollack, J. M., Finkel, E. J. (2013). Mind-sets matter: A meta-analytic review of implicit theories and self-regulation. *Psychological Bulletin*, 139(3), 655-701. doi:10.1037/a00295315.

Campitelli, G., & Gobet, F. (2011). Deliberate practice: Necessary but not sufficient. *Current Directions in Psychological Science*, 20(5), 280-285. doi:10.1177/0963721411421922

Castella, K., & Byrne, D. (2015). My intelligence may be more malleable than yours: The revised implicit theories of intelligence (self-theory) scale is a better predictor of achievement, motivation, and student disengagement. *European Journal of Psychology of Education*, 30(3), 245-267. doi:10.1007/s10212-015-0244-y

Chookittikul, J., & Chookittikul, W. (2008). Six sigma quality improvement methods for creating and revising computer science degree programs and curricula. *2008 38th Annual Frontiers in Education Conference*, 2008, F2E-15-F2E-20. doi:10.1109/FIE.2008.4720541

Cohen J. (1988). Statistical Power Analysis for the Behavioral Sciences. New York, NY: Routledge Academic

Cutts, Q. Cutts, E. Draper, S., O'Donnell, P. & Saffrey, P. (2010). Manipulating Mindset to Positively influence Introductory Programming Performance. In K. Bala (Ed.), DBLP Conference: Proceedings of the 41st ACM technical symposium on Computer science education (pp. 431-435). New York, NY: ACM. SIGCSE. doi>10.1145/1734263.1734409

Diener, C. I., Dweck, C. S., Diener, C. I., & Dweck, C. S. (1978). An analysis of learned helplessness: Continuous changes in performance, strategy, and achievement

cognitions following failure. *Journal of Personality and Social Psychology*, 36(5), 451-462. doi:10.1037/0022-3514.36.5.451

Diener, C. I., & Dweck, C. S. (1980). An analysis of learned helplessness: II. The processing of success. *Journal of Personality and Social Psychology,* 39(5), 940952. doi:10.1037/00223514.39.5.940

Doerschuk, P., Juarez, V., Jiangjiang Liu , , Vincent, D., Doss, K., & Mann, J. (2013). Introducing programming concepts through video game creation. *2013 IEEE Frontiers in Education Conference (FIE),* , 523-529. doi:10.1109/FIE.2013.6684879

Dweck, C. S. (1999). *Self-theories: their role in motivation, personality, and development*. London: Routledge.

Dweck, C. S. (2002). The development of ability conceptions. In A. Wigfield & J. Eccles (Eds.), *The development of achievement motivation*. New York, NY: Academic Press.

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

Dweck, C. S. (2011). Implicit theories. In P. Van Lange, A. Kruglanski, & E. T. Higgins (Eds.), *Handbook of theories in social psychology.* Vol 2. (pp. 43–61). Los Angeles, CA: Sage.

Dweck, C. S., Goetz, T. E., Strauss, N. L., Dweck, C. S., Goetz, T. E., & Strauss, N. L. (1980). Sex differences in learned helplessness: IV. An experimental and naturalistic study of failure generalization and its mediators. *Journal of Personality and Social Psychology*, 38(3), 441-452. doi:10.1037/0022-3514.38.3.441

Dweck, C. S., & Leggett, E. L. (1988). A Social–Cognitive approach to motivation and personality. *Psychological Review*, 95(2), 256273. doi:10.1037/0033295X.95.2.256

Dweck, C. S., & Master, A. (2005). Self-theories motivate self-regulated learning. In B. J. Zimmerman & D. H. Schunk (Eds). *Motivation and self-regulated learning: Theory, research, and applications.* (pp 31-51). New York, NY: Lawrence Erlbaum Associates.

Dweck, C. S., & Molden, D. C. (2005). Self-theories. In Elliot, A. J., & Dweck, C. S. (Eds.). *The handbook of competence and motivation*. (pp 122-143). New York, NY: Guilford.

Dweck C. S., & Sorich, L. A. (1999). Mastery-Oriented Thinking. In Snyder, C. R. (Eds.). *Coping: The psychology of what works*. (pp. 232-251). New York: Oxford University Press.

Henderson, V, L., & Dweck, C. S (1990). Motivation and Achievement. In Feldman, S. S., & Elliott, G. R (Eds). *At the threshold: The developing adolescent*. (pp 308-329). Cambridge, Mass: Harvard University Press.

Elliott, E. S., & Dweck, C. S. (1988). Goals: An approach to motivation and achievement. *Journal of Personality and Social Psychology*, 54(1), 5-12. doi:10.1037/0022-3514.54.1.5

Elliot, A. J. (2005). A conceptual history of the achievement goal construct. In Elliot, A. J., & Dweck, C. S. (Eds.). *The handbook of competence and motivation*. (pp 122-143). New York, NY: Guilford.

Erdley, C. A., Cain, K. M., Loomis, C. C., Dumas-Hines, F., & Dweck, C. S. (1997). Relations among children's social goals, implicit personality theories, and responses

to social failure. *Developmental Psychology*, 33(2), 263-272. doi:10.1037/0012-1649.33.2.263

Evans, J. D. (1996). Straightforward statistics for the behavioral sciences. Pacific Grove, CA: Brooks/Cole Publishing

Field, A. P. (2009). *Discovering statistics using SPSS: And sex drugs and rock 'n' roll* (3rd ed.). Los Angeles: SAGE Publications.

Good, C., Aronson, J., & Inzlicht, M. (2003). Improving adolescents' standardized test performance: An intervention to reduce the effects of stereotype threat. *Journal of Applied Developmental Psychology*, 24(6), 645-662. doi:10.1016/j.appdev.2003.09.002

Grant, H., & Dweck, C. S. (2003). Clarifying achievement goals and their impact. *Journal of Personality and Social Psychology*, 85(3), 541-553. doi:10.1016/S0022-3514(03)02907-8

Guzdial, M. (2011). From science to engineering. *Commun. ACM*, vol. 54, no. 2, pp. 37–39.

Higgins, E. T., Kruglanski, A. W., & Lange, Paul A. M. Van. (2012). *Handbook of theories of social psychology*. Vol. I and II. Los Angeles: SAGE.

Hong, Y., Chiu, C., Lin, D. M., Wan, W., & Dweck, C. S. (1999). Implicit theories, attributions, and coping: A meaning system approach. *Journal of Personality and Social Psychology*, 77(3), 588-599. doi:10.1037/0022-3514.77.3.588

Hoyt, C., Burnette, J., & Innella, A. (2012). I can do that: The impact of implicit theories on leadership role model effectiveness. *Personality and Social Psychology Bulletin, 38*(2), 257-268. doi: 10.1177/0146167211427922

Hu, X., Chen, Y., & Tian, B. (2016). Feeling better about self after receiving negative feedback: When the sense that ability can be improved is activated. *The Journal of Psychology*, 150(1), 72-87. doi:10.1080/00223980.2015.1004299

Jenkins, T. (2002). On the Difficulty of Learning to Program. Leeds: University of Leeds.

King, R. B., McInerney, D. M., & Watkins, D. A. (2012). How you think about your intelligence determines how you feel in school: The role of theories of intelligence on academic emotions. *Learning and Individual Differences*, 22(6), 814-819. doi:10.1016/j.lindif.2012.04.005

Kinnunen, P., & Simon, B. (2012). My program is ok – am I? computing freshmen's experiences of doing programming assignments. *Computer Science Education*, 22(1), 1-28. doi:10.1080/08993408.2012.655091

Kurland, D. M., Pea, R, D., Clement, C., & Mawby, R. (1989). A study of the Development of Programming ability and thinking skills in the high school students. In E. Soloway, & J. C. Spohrer (Eds.), *Studying the novice programmer* (pp 83-112). Hillsdale, N.J: L. Erlbaum Associates.

Luo, W., Lee, K., Ng, P. T., & Ong, J. X. W. (2014). Incremental beliefs of ability, achievement emotions and learning of Singapore students. *Educational Psychology*, 34(5), 619-634. doi:10.1080/01443410.2014.909008

McCroskey, J. C., & Richmond, V. P. (1989). Bipolar scales. In P. Emmert, & L. L. Barker (Eds.), *Measurement of communication behavior* (pp. 154-167). White Plains, NY: Longman, Inc.

Martinez, M. E. (2010). *Learning and cognition: The design of the mind*. Upper Saddle River, N.J: Merrill.

Murphy, L., & Thomas, L. (2008). Dangers of a fixed mindset. *ACM SIGCSE Bulletin*, 40(3), 271-275. doi:10.1145/1597849.1384344

(N/A) (N/A). Example Calculation of Eta-squared from Mixed ANOVA SPSS Output. Retrieved from http://wilderdom.com/research/ExampleCalculationOfEta-SquaredFromSPSSOutput_Mixed.pdf

Nussbaum, A., & Dweck, C. (2008). Defensiveness versus remediation: Self-theories and modes of self-esteem maintenance. *Personality and Social Psychology Bulletin,* 34(5), 599612. doi:10.1177/0146167207312960

Ommundsen, Y. (2003). Implicit theories of ability and self-regulation strategies in physical education classes. *Educational Psychology,* 23(2), 141-157. doi:10.1080/01443410303224

Paunesku, D., Walton, G. M., Romero, C., Smith, E. N., Yeager, D. S., & Dweck, C. S. (2015). Mind-set interventions are a scalable treatment for academic underachievement. *Psychological Science*, 26(6), 784-793. doi:10.1177/0956797615571017

Perkings, D. N., Hancock, C., Hobbs, R., Martin F., & Simmons, R. (1986). Conditions of Learning in Novice Programmers. In Soloway, E & Spoherer. J. C. (Eds), *Studing the Novice Programmer* (pp. 261-279). Hillsdale, N.J. Lawrance Erlabum Associates.

Robins, R. W., & Pals, J. L. (2002). Implicit self-theories in the academic domain: Implications for goal orientation, attributions, affect, and self-esteem change. *Self and Identity*, 1(4), 313-336. doi:10.1080/15298860290106805

Roggerson, C. (2010). The fear factor: How it affects students learning to program in a tertiary environment(Vol. 9). Cape Town: University of Cape Town

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), 169-174. doi:10.1109/TE.2013.2288700

Schweizer, K., & Schreiner, M. (2010). Avoiding the effect of item wording by means of bipolar instead of unipolar items: An application to social optimism. *European Journal of Personality*, 24(2), 137150. doi:10.1002/per.748

Siddiqi, R., Harrison, C. J., & Siddiqi, R. (2010). Improving teaching and learning through automated short-answer marking. *IEEE Transactions on Learning Technologies*, 3(3), 237-249. doi:10.1109/TLT.2010.4

Simon, B., Hanks, B., Murphy, L., Fitzgerald, S., McCauley, R., Thomas, L., & Zander, C. (2008). Saying isn't necessarily believing: Influencing theories in Computing. Sydney: ACM.

Tillmann, N., de Halleux, J., & Tao Xie , . (2011). Pex4Fun: Teaching and learning computer science via social gaming. *2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T),* , 546-548. doi:10.1109/CSEET.2011.5876146

Tempelaar, D., Rienties, B., Giesbers, B., & Gijselaers, W. (2015). The pivotal role of effort beliefs in mediating implicit theories of intelligence and achievement goals and academic motivations. *Social Psychology of Education*, 18(1), 101120. doi:10.1007/s1121801492817

Van Gog, T., & Kester, L. (2012). A test of the testing effect: Acquiring Problem‐Solving skills from worked examples. *Cognitive Science*, 36(8), 1532-1541. doi:10.1111/cogs.12002

Wigfield, A., & Eccles, J. S. (2002). *Development of achievement motivation*. San Diego: Academic Press.

Wigfield, A., & Wentzel, K. R. (2009). *Handbook of motivation at school*. New York;London;: Routledge.

Yeager, D. S., Trzesniewski, K. H., & Dweck, C. S. (2013). An implicit theories of personality intervention reduces adolescent aggression in response to victimization and exclusion. *Child Development*, 84(3), 970-988. doi:10.1111/cdev.12003

Yeager, D. S., Johnson, R., Spitzer, B. J., Trzesniewski, K. H., Powers, & J., Dweck, C. S. (2014). The far-reaching effects of believing people can change: Implicit theories of personality shape stress, health, and achievement during adolescence. *Journal of Personality and Social Psychology*, 106(6), 867-884. doi:10.1037/a0036335

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

Zhao, Q., Zhang, J., & Vance, K. (2013). Motivated or paralyzed? Individuals' beliefs about intelligence influence performance outcome of expecting rapid feedback. *Learning and Individual Differences*, 23(Complete), 168-171. doi:10.1016/j.lindif.2012.07.019

Zimmerman, B. J., & Schunk, D. H. (2008). *Motivation and self-regulated learning: Theory, research, and applications*. New York: Lawrence Erlbaum Associates.

Appendix A

Implicit theories effect on academic outcomes

Related factor influenced	Implicit theory or mindset	
	Entity	Incremental
Challenges	Avoid	Embrace
Obstacles	Not persist	Keep persisting
Goal orientation	Mastery goals	Performance goals
Effort	Negative value; avoid	Positive value; embrace
Feedback	Avoid; discard	Pay attention, see as path to growth
Success of others	Feel threatened	See as role models to follow
As a result	Achieve higher success	Achieve lower success

Appendix B

Online Survey

Demographic questions

Please select your major
- Cognitive Science
- Computer Science
- Biology
- Chemistry
- History
- Other, please specify _____

What year are you in at Carleton?
- First
- Second
- Third
- Fourth
- Other, please specify _____

Please select your gender
- Male
- Female
- Prefer not to answer

Please specify your age

Please select your ethnicity
- Caucasian/European descent
- Aboriginal (North American Indian, Métis or Inuit)?
- Arab (e.g., Saudi, Egyptian, Iraqi, Lebanese, Palestinian, Syrian etc.)
- Black (e.g., African, African American, African Canadian, Caribbean)
- East Asian (e.g., Chinese, Japanese, Korean, Polynesian)?
- Latin American?
- South Asian (e.g. Indian, Pakistani, Sri Lankan, Bangladeshi)
- Southeast Asian (e.g. Burmese, Cambodian, Filipino, Indonesian, Laotian, Malaysian, Thai, Vietnamese)?
- West Asian (Afghan, Armenian, Iranian, Israeli, Turks etc.)
- Other: _____
- Prefer not to answer

Are you a native English speaker?
- Yes
- No
- Prefer not to answer

Appendix B

Online Survey

Prior programming experience questions

How much experience do you have with computer programming?

- ❍ None (never done it)
- ❍ Some (Took a course in high school but haven't done since them)
- ❍ A lot (took two or more courses and/or write programs for fun

Have you ever used any visual computer programming tool?

- ❍ Yes, please specify _____
- ❍ No

Have you ever used Snap!?

- ❍ Yes
- ❍ No

Appendix B

Online Survey

Items from the student questionnaire, in blue incremental items, in purple entity items

General Beliefs
You have a certain amount of intelligence, and you can't really do much to change it
No matter how much intelligence you have, you can always change it quite a bit
Your intelligence is something about you that you can't change very much
No matter who you are, you can significantly change your intelligence level
You can learn new things, but you can't really change your basic intelligence
You can change even your basic intelligence level considerably
To be honest, you can't really change how intelligent you are
You can always substantially change how intelligent you are
Programming Beliefs
You have a certain amount of computer programming ability and you can't really do much to change it
No matter how much computer programming ability you have, you can always change it quite a bit
Your computer programming ability is something about you that you can't change very much
No matter who you are, you can significantly change your computer programming ability
You can learn new things , but you can't really change your basic computer programming ability
You can change even your basic computer programming ability considerably
To be honest, you can't really change how much computer programming ability you have
You can always substantially change how much computer programming ability you have

Appendix B

Online Survey

Items from the student questionnaire, in blue incremental items, in purple entity items

Learning Goals
An important reason why I do my school work is because I like to learn new things
I like school work best when it makes me think hard
I like school work that I'll learn from even if I make a lot of mistakes
Performance Approach Goals.
I like school work best when I can do it perfectly without any mistakes
The main thing I want when I do my school work is to show how good I am at it
I like school work best when I can do it really well without too much trouble
Performance Avoidance Goals.
It's very important to me that I don't look stupid in class
An important reason I do my schoolwork is so I won't embarrass myself
An important reason I do my work for class is so others won't think I'm dumb
Effort Beliefs
The harder you work at something, the better you will be at it
When something is hard, it just makes me want to work more on it, not less
If you don't work hard and put in a lot of effort, you probably won't do well
If an assignment is hard, it means I'll probably learn a lot doing it
To tell the truth, when I work hard at my schoolwork, it makes me feel like I'm not very smart
It doesn't matter how hard you work -if you're not smart, you won't do well
If you 're not good at a subject, working hard won't make you good at it
If a subject is hard for me, it means I probably won't be able to do really well at it
If you're not doing well at something, it's better to try something easier

Appendix C

Thank you e-mail

Dear participant,

Thank you for your interest in participating in laboratory session of the study titled "Investigating Beliefs in Ability".

Because we are looking for participants with a certain response profile, at this time you are not eligible for participating in the next phase - but we will be in touch if anything changes.

Have a great day!

Appendix D

Follow-up instructions e-mail

Dear participant,

Thank you for filling in the survey – you qualify for the laboratory session, held in a research lab at Carleton University (VSIM building).

The study takes 2 hours and compensation is $20. In the study, you will be asked to read a brief text, fill in a questionnaire and do some problem solving activities. No prior experience is required.

Available slots are in the following google calendar: Research Calendar - Investigating Beliefs in
Ability. https://calendar.google.com/calendar/embed?src=calendar.iba%40gmail.com&ctz=America/Toronto

Please send your top two preferences in that order (day, time)
to jakelinegcelisrangel@cmail.carleton.ca . The researcher will confirm the date and send you directions to the research lab.

Should you have any comments or questions, please do not hesitate to contact the researcher at: jakelinegcelisrangel@cmail.carleton.ca

Thank you again for your interest!
The Investigating Beliefs in Ability Team

Appendix E

Research paper for control condition

Habits You can Use to Study More Efficiently[1]

Research Shows new Ways Students can Improve their Grades

The key to becoming an effective student is learning how to study smarter, not harder. This becomes more and truer as you advance in your education. An hour or two of studying a day is usually sufficient to make it through high school with satisfactory grades, but when college arrives, there aren't enough hours in the day to get all your studying in if you don't know how to study effectively.

[1] http://www.educationcorner.com/habits-of-successful-students.html, https://www.stonebridge.uk.com/blog/study-tips/study-tips-for-winter

While some students are able to breeze through school with minimal effort, this is the exception. The vast majority of successful students achieve their success by developing and applying effective study habits. The following are the top 20 study habits employed by highly successful students. So if you want to become a successful student, don't get discouraged, don't give up, just work to develop each of the study habits below and you'll see your grades go up.

1. Don't attempt to cram all your studying into one session.

Ever find yourself up late at night expending more energy trying to keep your eyelids open than you are studying? If so, it's time for a change. Successful students typically space their work out over shorter periods of time and rarely try to cram all of their studying into just one or two sessions. If you want to become a successful student then you need to learn to be consistent in your studies and to have regular, yet shorter, study periods.

2. Plan when you're going to study.

Successful students schedule specific times throughout the week when they are going to study -- and then they stick with their schedule. Students who study sporadically and whimsically typically do not perform as well as students who have a set study schedule. Even if you're all caught up with your studies, creating a weekly routine, where you set aside a period of time a few days a week, to review your courses will ensure you develop habits that will enable you to succeed in your education long term.

3. Study at the same time.

Not only is it important that you plan when you're going to study, it's important you create a consistent, daily study routine. When you study at the same time each day and each week, your studying will become a regular part of your life. You'll be mentally and emotionally more prepared for each study session and each study session will become more productive. If you have to change your schedule from time to time due to unexpected events, that's okay, but get back on your routine as soon as the event has passed.

4. Each study time should have a specific goal.

Simply studying without direction is not effective. You need to know exactly what you need to accomplish during each study session. Before you start studying, set a study session goal that supports your overall academic goal (i.e. memorize 30 vocabulary words in order to ace the vocabulary section on an upcoming Spanish test.)

5. Never procrastinate your planned study session.

It's very easy, and common, to put off your study session because of lack of interest in the subject, because you have other things you need to get done, or just because the assignment is hard. Successful students do not procrastinate studying. If you procrastinate your study session, your studying will become much less effective and you may not get everything accomplished that you need to. Procrastination also leads to rushing, and

rushing is the number one cause of errors.

6. Start with the most difficult subject first.

As your most difficult assignment or subject will require the most effort and mental energy, you should start with it first. Once you've completed the most difficult work, it will be much easier to complete the rest of your work. Believe it or not, starting with the most difficult subject will greatly improve the effectiveness of your study sessions, and your academic performance.

7. Always review your notes before starting an assignment.

Obviously, before you can review your notes you must first have notes to review. Always make sure to take good notes in class. Before you start each study session, and before you start a particular assignment, review your notes thoroughly to make sure you know how to complete the assignment correctly. Reviewing your notes before each study session will help you remember important subject matter learned during the day, and

make sure studying targeted and effective.

8. Make sure you're not distracted while you're studying.

Everyone gets distracted by something. Maybe it's the TV. Or your family. Or maybe it's too quite. Some people actually study better with a little background noise. When you're distracted while you're studying you (1) lose your train of thought and (2) you're unable to focus -- both of which will lead to very ineffective studying. Before you start studying find a place where you won't be disturbed or distracted. Some people this is a quite cubical in the recesses of the library.

9. Use study groups effectively.

Ever heard the phrase "two heads are better than one"? Well this can be especially true when it comes to studying. Working in groups enables you to (1) get help from others when you're struggling to understand a concept, (2) complete assignments more quickly, and (3) teach others whereby helping both the other students and your-self to internalize the subject matter. However, study groups can become very ineffective if they're not structured and if groups members come unprepared. Effective students use study

groups effectively.

10. Review your notes, schoolwork and other class materials over the weekend.

Successful students review what they've learned during the week over the weekend. This way they're well prepared to continue learning new concepts that build upon previous coursework and knowledge acquired the previous week.

11. Tell a tale.

Turning the details you need to remember into a crazy story helps make the information more meaningful. For example, if you need to remember the order of some mathematic operations such as "PEMDAS", try this way: Philip (P) wanted to eat (E) his friend Mary (M) but he died (D) from arsenic (AS) poisoning.

12. Move around.

Research suggests studying the same stuff in a different place every day makes us less likely to forget that information. Every time we move around (from the library to the coffee shop or the coffee shop to the toilet seat), we are study the same material, so it will be more likely to remember it under different circumstances, for instance, during the exam.

13. Put yourself to the test.

Quizzing ourselves may be one of the best ways to prepare for the real deal. And don't worry about breaking a sweat while trying to remember the name of the 37th U.S. president (fyi, it's Nixon): The harder it is to remember a piece of information in practice mode, the more likely we are to remember it in the future.

14. Write it out.

Put those third-grade penmanship lessons to good use. Research suggests we store information more securely when we write it out by hand than when we type it. Start by recopying the most important notes from the semester onto a new sheet of paper.

15. Caffeine and alertness

Hit the local coffee shop for something caffeine filled (with limits of course!); there's lots of research suggesting coffee (and tea) keeps us alert, especially when nothing seems more exciting than the shiny gum wrapper on the library floor.

16. Treat yourself!

A healthy holiday cookie, a walk around the block, five minutes on Twitter—whatever floats your boat. Knowing there's a little reward waiting for us at the end of just a few pages makes it easier to beat procrastination while slogging through a semester's worth of notes.

17. Take a time out.

Taking time to plan is one of the most important skills a student can have. Don't just start the week with the vague goal of studying for a history exam—instead, break up that goal into smaller tasks. Pencil it in on the calendar like a regular class: For example, allot every day from 1 to 3 p.m. to review 50 years' worth of info.

18. Say "Om."

Just before staring at a piece of paper for three hours, stare at a wall for three minutes. Research suggests meditation can reduce anxiety and boost attention span. While those studies focus mostly on regular meditation, there's no harm in trying it out for a few minutes to calm pre-test jitters.

19. Nix the 'net.

We've all been there, facing the siren call of a friend's Facebook wall on the eve of a giant exam. If a computer is necessary for studying, try an app that blocks the Internet for a short period of time and see how much more you get done.

20. Learn what works.

Some people are early birds, some are night owls; some prefer to study with a pal, others need complete and total silence. Experiment to find what's most effective for you, and then stick with it!

We're confident that if you'll develop the habits outlined above you'll see an improvement in your academic success.

Appendix E

Research papers for experimental condition

You Can Increase Your Intelligence[1]

Research Shows the Brain Can Be Developed Like a Muscle

When individuals think about intelligence, many believe that a person is born with a certain amount of intelligence or ability and stays that way for life. To illustrate, how do you feel about your ability to do a challenging activity like math or computer programming? Do you believe you can learn these topics? Or do you instead feel that you just aren't very good at them and that can't be changed? In fact, research shows that the brain is like a muscle—it changes and gets stronger when people use it and learn new things. This especially applies to challenging topics that we are not initially good at – **so anyone can improve their ability, even on topics they struggle with**. Let's look at some evidence for that claim.

[1] based in part on https://www.mindsetworks.com/websitemedia/youcangrowyourintelligence.pdf

---

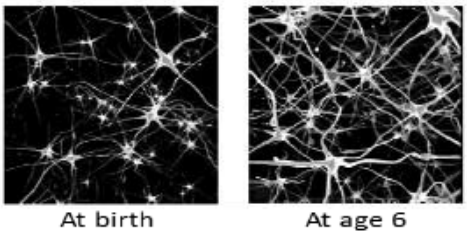
Everyone knows that when people lift weights, their muscles get bigger and people get stronger. A person who can't lift 20 pounds when they first start exercising can get strong enough to lift 100 pounds after working out for a long time. That's because the muscles become larger and stronger with exercise. At the same time, when people stop exercising, their muscles shrink and get weaker. That's why people say "Use it or lose it!"

What about cognitive tasks – like math or computer programming – can we improve our ability for these? In short, yes. There is evidence that when we invest effort and time, brain regions change and become *better connected*, like muscles do when they are exercised. In fact, the communication among neurons (brain cells) is what allows people to think, solve problems, and master new activities. The more and better connections between the neurons, the higher one's ability becomes – and as already pointed out, these connections can be developed with effort.

Brain Plasticity, Learning and Intelligence

This brain's ability to change as a result of practice and learning is called plasticity or neuroplasticity. Thus, in addition to genetic factors, *the effort invested* plays a significant role in brain plasticity. So, as we said earlier, challenging experiences can stimulate certain areas of brain to increase the connectivity of its neurons, allowing people to *increase their ability* (i.e., intelligence).

How does age interact with plasticity? You may have already heard that when we are very young, are brains are plastic (able to form new connections). Newborn babies' neurons have few neurons and few connections among them, but in their first years, their brains develop many new neurons and connections – for instance, toddlers have about 1,000 trillion connections. At this stage, the brain starts to optimize these connections: those that are most used get stronger, whereas those not used will start to fade.



At birth At age 6

Figure 1. Growth of neuron connections in a child from birth to 6 years old

For adults, fairly recently the general assumption was that the human brain did not generate new neurons. However, in the 70's, the use of novel methods allowed researchers to identify that in fact neurons were generated in adulthood. For example, Eriksson et al. (1998) yielded considerable evidence that the mammalian brain, including that of humans, generates new neurons in the hippocampus, and that these were critical for peoples' ability to learn. On average, the hippocampus of a healthy adult can generate around 700 new brain cells (neurons) every day – and these neurons can travel to other areas of the brain, supporting all kinds of cognitive skills. Furthermore, Gould et al (1999) showed that this daily amount of new neurons in the hippocampus could be enhanced when one learns new skills or concepts. However, these new cells will not survive unless they *are used*, for instance, in the reinforcement of the learning processes. These changes associated with learning occur mostly at the level of the connections between neurons, this is at dendritic level, where new connections can be formed and the internal structure of the existing synapses and dendritic ramifications can change (see Figure 2).

Figure 2. Dendritic plasticity. Changes involving both the extension and retraction of particular dendritic branches.

This explains why initially, something like math or programming can be challenging, i.e., because we do not yet have the right connections between our neurons. If we are willing to persevere through challenges and practice the new skill, connections will form, and eventually the task will become easier and more automatic.

What did these early discoveries in the neurosciences show? They showed us that brains are indeed plastic and they can form new connections. These connections are critical because they are responsible for allowing us to improve our ability – to grow our intelligence! Let's look at some evidence of that.

Woollett and Maguire (2016) followed taxi driver candidates' as they were trained to become licensed cab drivers. Over four years, when the candidates learned the complex layout of roughly 25,000 irregular London's streets, researchers recorded the grey matter intensity in the hippocampal area from the would-be taxi drivers before and after training. At the end of four years, researchers compared the candidates who qualified the exam, to two other groups (a control and those who failed). Their results showed that hippocampal volume in the grey matter of the brain significantly increased in the candidates who passed, but those who failed the exam, and in the control group, showed no changes (see Figure 3).

Interestingly, in other study by Woollett, Spiers, and Maguire, (2009) the opposite pattern of hippocampal gray matter volume was observed in retired taxi drivers, where these brain areas reduced their in volume, hinting that any changes acquired through learning might be reversed when one stops applying that cognitive skill.

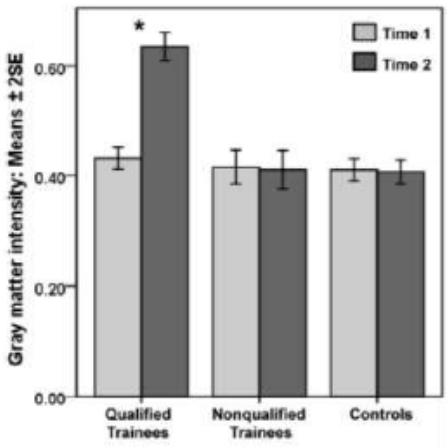


Figure 3. Grey matter intensity as a function of trainees and controls.

This neuroplasticity capability can also be observed in the brains of bilinguals (Mechelli et al., 2004). Specifically, the left inferior parietal cortex is larger in bilingual brains than in monolingual brains. What is even more interesting is that even though most people think that learning a second language once they are adult is very hard, new research have found the opposite. Using PET scans to measure the activity in the brain, researchers found that when people did certain special exercises, areas of their brain that they had never used before became active and this improved their ability to learn the new language. Thus, it is never too late to learn something new!

The sceptical reader may be thinking that anyone can learn a language, but not everyone can do certain academic topics. So what evidence do we have about ability related to challenging academic topics like math? There is a wealth of evidence that students who worked hard to improve their ability for topics like math did in fact improve their ability (Blackwell et al. 2015) – this was measured by improved grades (one study reported an average grade increase of a full letter grade), as well as other factors. The interesting thing is that some of the students initially thought they couldn't learn the topic because they weren't inherently good at it. This was not the case – effort was the key predictor of success.

All these examples demonstrate that neuroplasticity means that intelligence is malleable and improvable through practice and hard work – even for challenging topics that one initially struggles with. Put another way, ability for a given topic is not something we are born with – it is something we develop.

The key to Growing the brain: Practice!

Hence, just like the weightlifter, one can be a brain athlete by learning new things and practicing new skills. However, many people miss out the chance to grow a "stronger" brain because they think they cannot do it, or that it is too hard. When that happens, we have to remember that getting better at something is a matter of practice.

Improving ability in academic contexts

No one thinks toddlers are stupid because they can't read. They just haven't learned how to yet. But some people will label themselves or another student as not very smart (not a nice label!) if they struggle with a topic like math or programming. We now know that it isn't their lack of ability that makes them struggle, but lack of practice. Think back to the early years in elementary school – was there a student in your class who already knew how to manipulate fractions in math class and if so did you think they were "good at math"? What if you found out that their mom was a math professor and the student was good because they enjoyed working with their mom on the math exercises (and so they practised a lot)! The key here to realize is that anyone could learn to do as well if they practice. Remember that the only difference between the "top" student and those other students is how much

they the former have practiced and build their brain connections.

Summing it up: New Ways to Think About Ability and Intelligence

1. Keep in mind the relationship between learning and "brain training." The brain is like a muscle that needs to be trained – training will in turn improve ability.

2. Forget about "naturally smart". People who are labeled as such are those who worked hard and practiced a lot in order to master the task they are thought to be "good" at.

3. Redefine "genius." The myth's been busted: even geniuses have to work hard!

4. Use the word "yet." Whenever you feel struggling with a task, just think that you have not mastered it yet. Practice, hard work and the right strategies will help you to master it.

5. Think realistically about time and effort. It takes time to learn.

6. Try different learning tactics. There is no one-size-fits-all model for learning. What works for one person may not work for you.

7. Pay attention to your own mistakes. Errors are signals that we can use as guides to know where we have not yet mastered a concept and what we need to improve.

8. Replace the word "failing" with the word "learning." When you make a mistake or fall short of a goal, although you did not reach your goal, you still can learn how to do things differently; you are still learning and generating new connections in your brain!

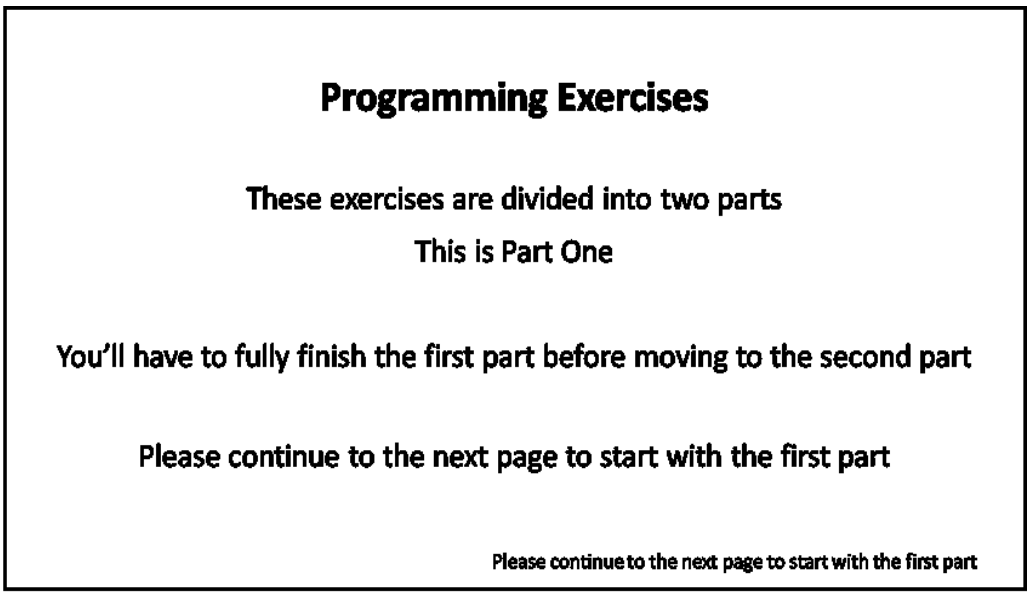
9. View challenges as opportunities. When you expose yourself to challenging concepts or tasks, you have a unique opportunity to learn more. Leave the easy things for later and take the risk to master inspiring tasks now!

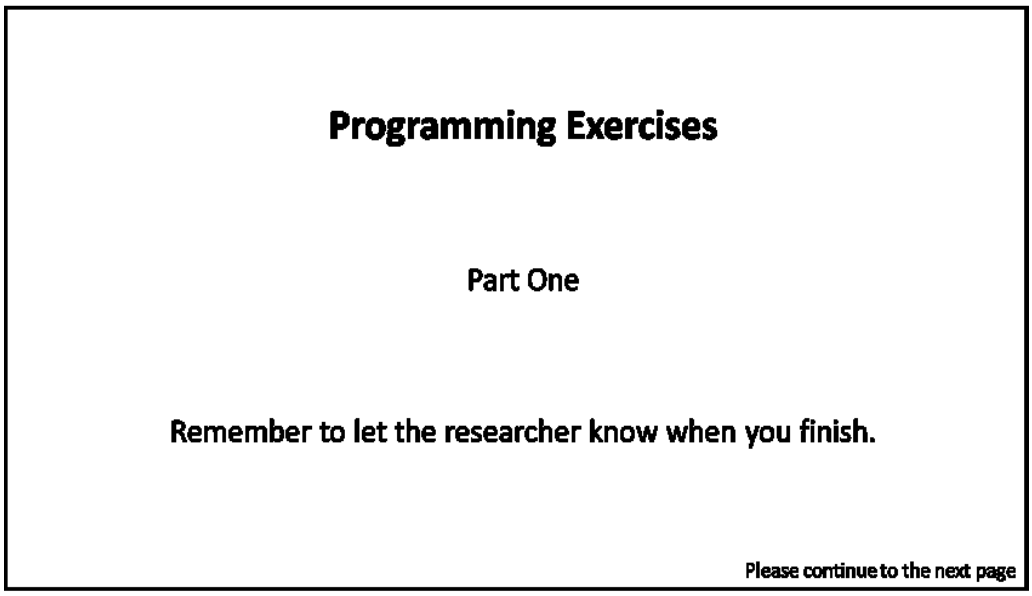
10. Re-interpret the meaning of negative feedback as a positive opportunity. Do not take criticism as a signal of your intelligence - take it as a chance to improve your-self!

11. Take ownership over your attitude. Once you are aware that everyone can become smarter through practice and hard work, own it. Acknowledge yourself as someone who possesses the ability to learn and be proud to let this new approach to guide you throughout your educational career.

Appendix F

Similar Exercise





Appendix F

Similar Exercise (cont.)

Exercise One

Please write in Snap, *exactly* the same Guessing Game you have been reviewing

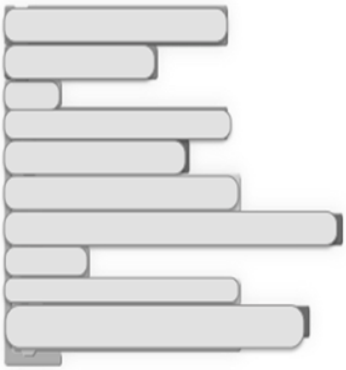
To get a hint on the correctness of your program, click **here**.

Once you finish this exercise, let the researcher know so she can move you to the second part

Exercise One

You will know your activity is right when your program:

1. Has a total of 10 lines and looks like the shape on the right
2. Creates the secret number variable only once
3. Keeps asking the user to guess the secret number until the user makes the right guess
4. Depending on whether the user's guess is right or wrong, the program will display a different message
5. The program ends when the user makes the right guess



To return to exercise one, click here.

Appendix F

Transfer Exercise

Programming Exercises

Part Two

Remember to let the researcher know when you finish.

Please continue to the next page to start with the second part

Part Two – Introduction

The **"age"** program shown on the Snap window stores a person's initial age, and then increments that age by 1 year. Please run it now.

The following exercises involve modifying this **"age"** program shown in your Snap window. In case you need to start from scratch, the code for the "age" program is on the left side of the next page, that page also has the code for the "Guessing Game" program from the last exercise (on the right side), since parts of it may be useful.

Please continue to the next page

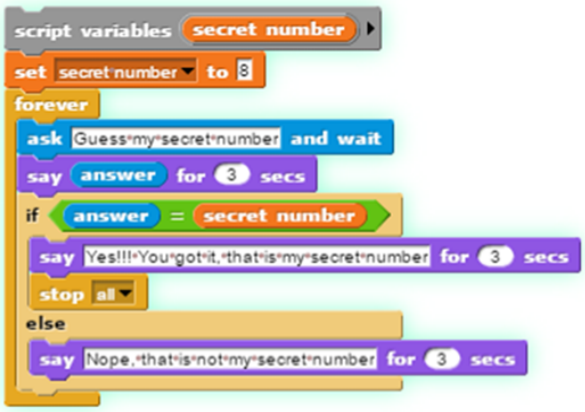
Appendix F

Transfer Exercise (cont.)

Part Two – Age and Guessing Game codes

Age Code

Guessing Game Code



Please continue to the next page

Part Two – The three exercises

Please modify the "age" program in these three ways:

Exercise One: The "age" program increments the variable age only once. Change the program so that it keeps incrementing the variable "age" forever.

Exercise Two: Modify your program from exercise one, so that it stops incrementing age once it becomes 100.

Exercise Three: Modify your program from exercise two, so that the first thing it does is ask for a user's name, and then have it displays that name each time it displays the age

Once you finish with these exercises, please let the researcher know.

Note. The transfer exercise was presented as three sub-exercises.

Appendix G

Participants' instructions

Participant's Instructions

The Programming Activities are divided in two parts. The first one –which is instructional, will last no more than 50 minutes and the second one –which are exercises, a maximum of 30 minutes.

To go through these programming activities just pretend that you are taking a programming class that has asked you to write a "Guessing Game" program.

This program will be written using a programming platform called Snap.

Snap is a visual programming language that lets you create programs by snapping blocks of instructions together.

To help you, we will provide 5 brief instructional videos that will explain how to write the "Guessing Game" using key programming topics (*variables*, *user input*, *loops*, *and conditionals*).

These 5 videos are in total, less than 20 minutes.

After watching each video, we ask you to practice the concepts presented in the video by replicating in Snap the same program as you saw in the video. When doing this mandatory post-video practice, you can also go back and forth in topics and videos watching them again, rewinding/forwarding etc.

However, it is up to you how much post-video practice you choose to do.

Just keep in mind that, for the instructional part, you will have up to 50 minutes to watch the videos and do your mandatory post-video practice. If you finish before, or feel ready to continue, please let the researcher know.

Once you finish the instructional part, (watching the videos and doing your mandatory post-video practice), you will be asked to do the exercises: writing your own program without any access to the videos. You will have an additional 30 minutes to do these exercises.

If you have any question or comments about the procedure, please ask the researcher now. During the programming session, she is not permitted to answer any questions related to the programming topics. However, in case you get in technical difficulties she can help you; for instance if the monitor screen becomes black.

Appendix H

Grading scheme for similar exercise

(A)	Declare a variable	0.50
(a.1)	if variable is not declared	-0.50
(B)	Initialize the variable	0.50
(b.1)	if variable is not initialized	-0.50
(b.1)	if variable is wrongly initialized	-0.25
(C)	Loop	0.50
(c.1)	if there is no loop	-0.50
(D)	Ask user to guess the secret number repeatedly	0.50
(d.1)	if no ask instruction is present	-0.50
(d.2)	if the ask is not inside the loop	-0.25
(E)	Say the secret number repeatedly; Say instruction, inside loop, answer oval	3.00
(e.1)	if there are no say instruction neither answer oval (inside or outside the loop)	-3.00
(e.2)	if the say instruction is not repeated	-1.00
(e.3)	if there is no answer oval	-1.00
(e.4)	if say and answer oval are wrongly used	-1.00
(e.5)	if say instruction is misplaced (i.e. before the ask instruction)	-1.00
(F)	If “answer = secret number” -right condition inside forever	3.00
(f.1)	if there is no conditional	-3.00
(f.3)	if there is a conditional but it is wrongly written; i.e., condition is not an "="	-1.00
(f.2)	if the conditional is not inside the loop	-1.00
(f.4)	if the equity is not between the answer oval and the secret number	-1.00
(G)	Positive message is present when the condition is true	1.00
(g.1)	There is no positive message when the condition is true	-1.00
(H)	Stop the loop “if the user guess the secret number” -stop	1.00
(h.1)	if there is no stop instruction	-1.00
(h.2)	stop instruction in the wrong place (not within the condition)	-0.50
(J)	Negative message is present when the condition is false (Else part)	1.00
(j.1)	there is no negative message when the condition is false	-1.00
	Total points	11.0

Appendix H

Grading scheme for transfer exercise

(A)	Declare a variable	0.50
(a.1)	if variable is not declared	-0.50
(B)	initialize the variable	0.50
(b.1)	if variable is not initialized	-0.50
(b.2)	if variable is wrongly initialized	-0.25
(b.3)	if variable is initialized more than once	-0.25
(D)	Loop	1.00
(d.1)	if there is no loop	-1.00
(E)	Increment the age by 1 repeatedly in a loop	1.00
(e.1)	if the logic of the increments makes the program crash	-1.00
(e.2)	if there are no increments	-1.00
(F)	Say the age after every increment (and keep the original say out of the loop)	2.00
(f.1)	if there is not any say instruction	-2.00
(f.2)	if the two "say age" instructions are inside loop	-1.00
(f.3)	if there is only one "say age"	-1.00
(f.4)	if the 2nd "say age" is wrongly written	-1.00
(C)	Ask for a person's name only once	2.00
(c.1)	if no ask instruction is present	-2.00
(c.2)	if the ask instruction is more than once	-1.00
(G)	Say the user's name repeatedly -Say instruction, inside loop, answer oval	3.00
(g.1)	if there are no say instruction neither answer oval (inside or outside the loop)	-3.00
(g.2)	if the say instruction is not repeated	-1.00
(g.3)	if there is no answer oval	-1.00
(g.4)	if say and answer oval are wrongly used	-2.00
(g.5)	if two "say-answer ovals" are inside the loop	-1.00
(H)	Stop the loop "if the age is 100 stop" -right condition inside forever	2.50
(h.1)	if there is no instruction	-2.50
(h.2)	if instruction is wrongly written	-1.00
(J)	Stop the loop "if the age is 100 stop" -stop	0.50
(j.1)	if there is no stop instruction	-0.50
	Total points	13.0

Appendix I

Means, standard deviation, and mean differences for the rest of the constructs evaluated

Variable	Experimental (N=24)					Control (N=23)				
	Pre		Post		Mean Difference Post vs	Pre		Post		Mean Difference Post vs
	M	*SD*	*M*	*SD*		*M*	*SD*	*M*	*SD*	
Performance approach goals	2.74	.93	2.64	1.06	-.10	2.86	.70	2.71	.79	-.12
Performance avoidance goals	3.64	1.00	3.89	1.30	.25	3.51	1.11	3.48	1.10	-.03
Mastery goals	2.85	.80	2.57	.85	-.28	2.91	.73	3.07	.98	.16
Helpless attributions	4.02	.94	4.14	1.10	.11	3.77	1.13	3.88	1.14	.11

Appendix J

Tables for effect sizes comparisons

Effect sizes related to the increase from pre to post in Cohen's *d*

Variable	Experimental	Control
Entity programming beliefs	1.05	1.13
Incremental programming beliefs	1.25	1.04
Entity general beliefs	.84	1.01
Incremental general beliefs	1.31	.93
Collapsed general beliefs from Blackwell et al. (2007)	.66	.07
Entity effort	1.01	1.07
Incremental effort	1.02	1.04
Collapsed effort beliefs from Blackwell et al. (2007)	.48	.25

Effect sizes related to the conditional vs. control pre-post difference in Cohen's *d*

Variable	*d*
Entity programming beliefs	.66
Incremental programming beliefs	.65
Entity general beliefs	.78
Incremental general beliefs	.87
Collapsed general beliefs from Yeager et al. (2014)	.39
Entity effort	.50
Incremental effort	.25

Appendix K

Interactions for change in belief constructs

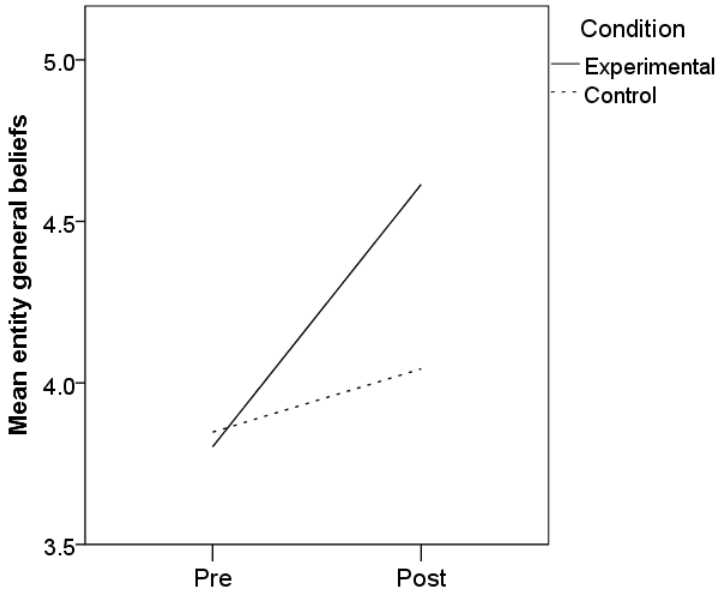


Figure 4.10 Interaction between time and condition for change in entity general beliefs

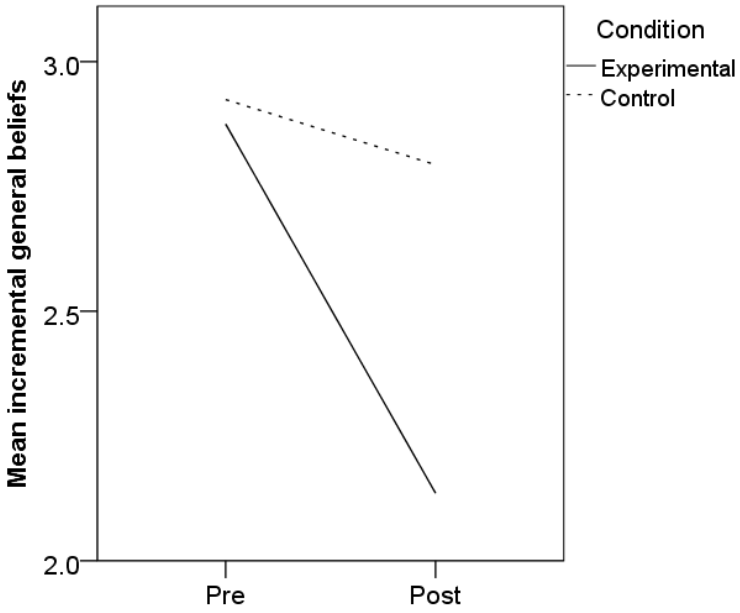


Figure 4.11 Interaction between time and condition for change in incremental general beliefs

Appendix K (cont.)

Interactions for change in belief constructs

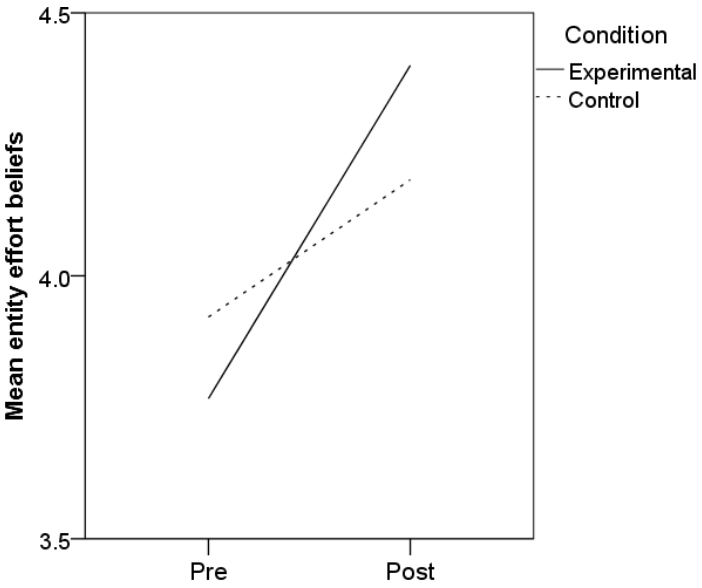


Figure 4.12 Interaction between time and condition for change in entity effort beliefs

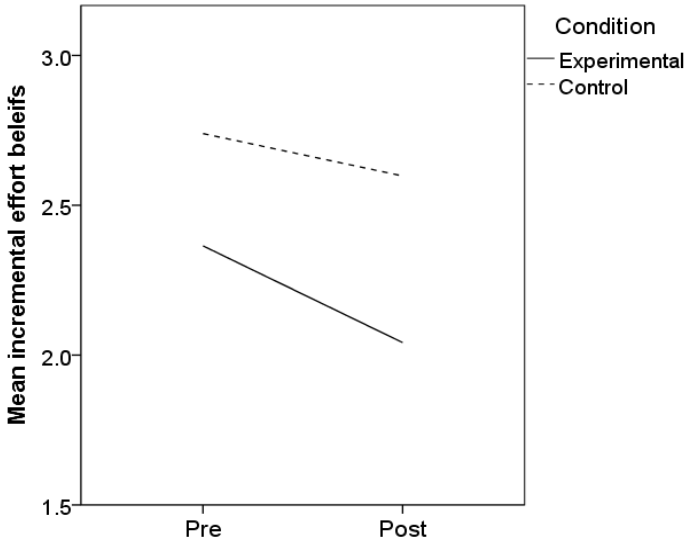


Figure 4.13 Interaction between time and condition for change in incremental effort beliefs