COGNITIVE, AFFECTIVE, AND DISPOSITIONAL COMPONENTS OF LEARNING $$\operatorname{PROGRAMMING}$

By

Alex Lishinski

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Educational Psychology and Educational Technology – Doctor of Philosophy

ABSTRACT

COGNITIVE, AFFECTIVE, AND DISPOSITIONAL COMPONENTS OF LEARNING PROGRAMMING

By

Alex Lishinski

Programming is a complex cognitive skill that develops over an extended period of time. The development of programming ability in introductory computer science courses has been found to be associated with a number of student factors, including cognitive, affective, and dispositional factors. Prior research on the student factors that are associated with success in programming has examined them in isolation from one another, and detailed pictures of the interactions of these factors over time are nonexistent. This dissertation focuses on building a detailed model of the factors that contribute to students' learning outcomes in programming.

This study collected data from 612 students enrolled in an introductory programming course on a number of cognitive, motivational, affective, and dispositional factors over the course of a semester. Students completed measures of problem solving ability, metacognitive self-regulation, self-efficacy, goal orientation, emotional responses, and personality traits. Path analysis and multiple regression were used to examine relationships between these student factors and course learning outcomes from exams and projects. The data analysis was used to determine how these student factors interact with one another and relate to student learning outcomes, and to determine which of these factors had the largest association with course outcomes. This study also examined the influence of students' emotional responses to programming projects. Structural equation models were used to determine whether there were reciprocal effects between emotional responses and project outcomes. Finally, this study also investigated the influence of students completing a programming self-evaluation task on their project scores and self-efficacy.

The results of this study demonstrated that problem solving ability was the biggest

predictor of students' performance on both project and exam scores, whereas conscientiousness and extraversion were significantly associated with project and exam scores respectively. When all factors were considered simultaneously, problem solving ability was the strongest significant predictor of both project and exam score outcomes, and conscientiousness also significantly predicted project scores. The results of this study also discovered reciprocal effects between students' self-efficacy beliefs and their project outcomes. Finally, the selfevaluation intervention provided evidence of a positive impact on students' project outcomes during the intervention period, but no evidence of impact on students' self-efficacy. These results provide CS education researchers with evidence on which student factors are related to programming learning outcomes, when cognitive, affective, and dispositional factors are considered together.

Copyright by ALEX LISHINSKI 2017 This dissertation is dedicated to Amy and Avery.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Aman Yadav for striking an excellent balance, letting me pursue what I was interested in while keeping me honest and giving me excellent guidance. I would also like to thank the other members of my committee. Jack Smith, for always being interested in hearing my ideas and asking thoughtful questions, Rich Enbody, for opening up his class to me and helping me make my study happen, and Matt Koehler, for stepping in to join my committee at the last minute, and for always giving me timely and extensive feedback on my writing. I would also like to thank my EPET friends and colleagues: Jon Good, Josh Rosenberg, Phil Sands, Sarah Gretter, Spencer Greenhalgh, and Chris Seals, for working with me and being good people. Finally I would like to thank my wife Amy and my son Avery. Amy, you did so much to help me finish this dissertation that it more than offset how much Avery made it harder.

TABLE OF CONTENTS

LIST OF TABLES if							
LIST OF FIGURES							
CHAPTEF 1.1 Ba 1.2 Ge	R 1 INTRODUCTION INTRODUCTION INTRODUCTION ackground Interview Interview oals of the Study Interview Interview	$\begin{array}{c} 1 \\ 1 \\ 2 \end{array}$					
CHAPTER 2.1 St 2.2 Co 2.3 2.3 M 2.3 2.3 2.3 2.3 2.3 2.3 2.3 2.3	R 2 LITERATURE REVIEW udent Factors that Influence Learning Outcomes	$\begin{array}{c} 4\\ 5\\ 6\\ 7\\ 8\\ 8\\ 9\\ 11\\ 11\\ 12\\ 13\\ 15\\ 16\\ 17\\ 18\\ \end{array}$					
CHAPTEH 3.1 Do 3.2 Sa 3.3 M 3.3 3.3 3.4 3.4 3.4 3.4 3.4 3.4	R 3 METHODOLOGYesign of the Studyumple of Studyeasures3.1 Problem Solving3.2 Motivated Strategies for Learning Scales (MSLQ)3.3 Big Five Personality Traits3.4 Course Learning Outcomes3.5 Data Collectionata Analysis4.1 IRT4.2 RQ1 Data Analysis4.3 RQ2 Data Analysis4.4 RQ3 Data Analysis4.5 RQ4 Data Analysis	21 21 22 25 27 28 30 30 30 30 30 33 36 37 38					
CHAPTER	R 4 RESULTS	40					

4.1	Descriptive St	atistics													40
4.2	RQ1 Results .														46
	4.2.1 Project	Score Path Ar	alysis .												48
	4.2.2 Exam	Score Path Ana	lysis												51
4.3	RQ2 Results .														55
4.4	RQ3 Results .														60
	4.4.1 Frustra	ted Questions													61
	4.4.2 Proud	Questions													65
	4.4.3 Inadeq	uate Questions													69
	4.4.4 Self-eff	icacy Questions													73
4.5	RQ4 Results .														78
СНАРТ	TER 5 DISCU	SSION			••		•••	••	• •	•	•••	•	• •	•	83
5.1	Discussion of a	cesults				• •				•					83
5.2	Implications .														93
5.3	Limitations .														94
	DICEC														~ -
APPEN	DICES		• • • • •	•••	••	•••	•••	•••	• •	•	•••	•	• •	•	97
APF	'ENDIX A	MSLQ Items	• • • • •	•••	••	•••	•••	•••	• •	•	•••	•	• •	•	98
APF	PENDIX B	Big 5 items .	• • • • •	•••	•••	•••	•••	•••	• •	•	•••	•	• •	• •	101
APF	PENDIX C	Self-Evaluation	Survey	• • •	••		•••	•••	• •	•	•••	•	• •	•	104
APF	PENDIX D	Problem Solvir	ng Items	•••	• •		•••	•••	• •	•	•••	•	• •	•	105
APF	PENDIX E	Emotional Res	ponse Qı	iestio	ns .		•••	•••	• •	•	•••	•	• •	•	112
APF	PENDIX F	Exam Example			••		•••	• •	• •	•	•••	•	• •	•	113
APF	PENDIX G	Programming J	Project E	Examp	ole .		•••		• •	•		•	• •	•	121
BIBLIC	GRAPHY				• •		•••			•		•	• •	•	130

LIST OF TABLES

Table 3.1:	Race/Ethnicity Breakdown	22
Table 3.2:	Top 20 Majors	23
Table 3.3:	MSLQ Reliability	26
Table 3.4:	Big Five Reliability	28
Table 4.1:	Project Descriptive Statistics	41
Table 4.2:	Exam Descriptive Statistics	41
Table 4.3:	MSLQ Descriptive Statistics	42
Table 4.4:	Big Five Descriptive Statistics	44
Table 4.5:	Problem Solving Descriptive Statistics	45
Table 4.6:	Frustration Descriptive Statistics	45
Table 4.7:	Inadequate Descriptive Statistics	46
Table 4.8:	Self-efficacy Descriptive Statistics	47
Table 4.9:	Proud Descriptive Statistics	47
Table 4.10:	Project Score Model Fit Statistics	48
Table 4.11:	Standardized Path Coefficients: Project Score Initial Model	48
Table 4.12:	Project Score Modified Model Fit Statistics	49
Table 4.13:	Project Score Model Likelihood Ratio Test	50
Table 4.14:	Standardized Path Coefficients: Project Score Modified Model	50
Table 4.15:	Exam Score Model Fit Statistics	51
Table 4.16:	Standardized Path Coefficients: Exam Score Initial Model	52
Table 4.17:	Exam Score Modified Model Fit Statistics	52

Table 4.18: Exam Score Model Likelihood Ratio Test	53
Table 4.19: Standardized Path Coefficients: Exam Score Modified Model \ldots	54
Table 4.20: Project Score MR Model Summary Statistics	55
Table 4.21: Project Score MR Model Coefficient Estimates	56
Table 4.22: Exam Score MR Model Summary Statistics	57
Table 4.23: Exam Score MR Model Coefficient Estimates	58
Table 4.24: Project Score MR Model Coefficient Estimates by Gender	59
Table 4.25: Exam Score MR Model Coefficient Estimates by Gender	60
Table 4.26: Frustration Model Fit Statistics	61
Table 4.27: Frustration Model Estimated Path Coefficients	62
Table 4.28: Frustration Model Estimated Covariances	62
Table 4.29: Frustration Modified Model Fit Statistics	63
Table 4.30: Frustration Model Likelihood Ratio Test	64
Table 4.31: Frustration Modified Model Parameter Estimates	64
Table 4.32: Frustration Modified Model Estimated Covariances	64
Table 4.33: Proud Model Fit Statistics	65
Table 4.34: Proud Model Estimated Path Coefficients	66
Table 4.35: Proud Model Estimated Covariances	66
Table 4.36: Proud Modified Model Fit Statistics	67
Table 4.37: Proud Model Likelihood Ratio Test	68
Table 4.38: Proud Modified Model Estimated Path Coefficients	68
Table 4.39: Proud Modified Model Estimated Covariances	68
Table 4.40: Inadequate Model Fit Statistics	69

Table 4.41: Inadequate Model Estimated Path Coefficients	70
Table 4.42: Inadequate Model Estimated Covariances	70
Table 4.43: Inadequate Modified Model Fit Statistics	71
Table 4.44: Inadequate Model Likelihood Ratio Test	72
Table 4.45: Inadequate Modified Model Estimated Path Coefficients	72
Table 4.46: Inadequate Modified Model Estimated Covariances	72
Table 4.47: Self-efficacy Model Fit Statistics	73
Table 4.48: Self-efficacy Model Estimated Path Coefficients	74
Table 4.49: Self-efficacy Model Estimated Covariances	74
Table 4.50: Self-efficacy Modified Model Fit Statistics	75
Table 4.51: Self-efficacy Model Likelihood Ratio Test	76
Table 4.52: Self-efficacy Modified Model Estimated Path Coefficients	76
Table 4.53: Self-efficacy Modified Model Estimated Covariances	76
Table 4.54: Intervention Completers vs Non-completers: Pre-intervention Mean Differences ferences	79
Table 4.55: Intervention Effects: Project Scores During the Intervention	80
Table 4.56: Intervention Effects: Self-efficacy Scores During the Intervention	81
Table 4.57: Intervention Effects: Project Scores after the Intervention	81
Table 4.58: Intervention Effects: Self-efficacy Scores after the Intervention	82

LIST OF FIGURES

Figure 3.1:	Data Collection Timeline	30
Figure 3.2:	Problem Solving Sum vs MAP scores	33
Figure 3.3:	Initial Path Model	35
Figure 4.1:	Distribution of Individual and Total Project Scores	42
Figure 4.2:	Exam Score Distributions	43
Figure 4.3:	Problem Solving MAP scores: Pretest and Posttest	46
Figure 4.4:	Modified Path Model: Project Scores	49
Figure 4.5:	Modified Path Model: Exam Scores	53
Figure 4.6:	Frustrated Initial Model	61
Figure 4.7:	Frustrated Modified Model	63
Figure 4.8:	Proud Initial Model	65
Figure 4.9:	Proud Modified Model	67
Figure 4.10:	Inadequate Initial Model	69
Figure 4.11:	Inadequate Modified Model	71
Figure 4.12:	Self-efficacy Initial Model	73
Figure 4.13:	Self-efficacy Modified Model	75

CHAPTER 1

INTRODUCTION

1.1 Background

The national importance being placed on computing education, such as the CSforAll initiative to expand computer science (CS) education to K-12 students, provides both an opportunity and a need to better understand how student come to learn CS concepts. These efforts at the state level (such as in Arkansas, Indiana, and Florida) and at the local level (such as in New York and Chicago) largely focus on teaching students to program. Prior work has suggested that successfully teaching programming is a difficult endeavor (Jenkins, 2002; Sleeman, 1986), as only about two thirds of introductory programming students at the undergraduate level pass their course (Bennedsen and Caspersen, 2007; Watson and Li, 2014). Furthermore, the research on best teaching practices in programming courses is far from cohesive. There is little consensus about which instructional practices best support student learning (Pears et al., 2007).

The existing research investigating how students learn to program can be divided into two strands: research that focuses on task demands, and research that focuses on student factors. The research on task demands has focused on what it is about the material that makes programming difficult to learn (e.g. Lopez et al., 2008; Rivers et al., 2016; Robins, 2010). This research has focused on developing models of the subject matter that explain why learning to program is difficult. For example, the learning edge momentum model explained the difficulty of learning programming by citing the hierarchical interdependence of the concepts (Robins, 2010).

The research on student factors has examined how factors like students' background, demographics, academic aptitude, and social and emotional factors influence learning outcomes in programming courses (Bergin and Reilly, 2005). This research has found that some factors (e.g. self-efficacy) strongly predict student outcomes in programming, whereas other factors (e.g. standardized test scores) are only weakly related to programming learning outcomes. Many individual factors have been examined repeatedly, such as previous programming experience, math ability, and spatial reasoning, whereas factors like personality traits and metacognitive self-regulation have been seldom examined. However, there is a lack of research that has examined how various student factors together influence learning to program.

1.2 Goals of the Study

The main goal of this study was to develop a model of how cognitive (e.g. problem solving), affective (e.g. self-efficacy), and dispositional (e.g. conscientiousness) factors interact to influence student learning outcomes in an introductory programming course. The existing research on student factors has typically examined a small number of factors, which prevents more robust conclusions from being drawn about the relative importance of students' various individual characteristics. By measuring and examining a variety of constructs together, this study developed a more comprehensive model of the effects of student factors on learning outcomes in introductory programming.

Another goal of this study was to address the dearth of existing research on affective factors in introductory programming. This study focused on two categories of affective student factors that previous research has suggested may be very important for explaining programming students' learning outcomes: self-regulated learning (SRL) and emotional responses. Self-regulated learning theory has a strong empirical foundation in explaining student learning outcomes across educational contexts, and some studies have suggested that SRL factors may be important in programming as well. Programming students' emotional responses have been studied previously in a set of in-depth qualitative studies (Kinnunen and Simon, 2010, 2011), but there has not previously been an attempt to measure these emotional responses and connect them to outcomes with a large group of students. By studying the affective characteristics and experiences of programming students, the results of this research could inform the refinement of teaching practices to be sensitive to the influence of these factors.

The overall goal for this study was to provide a broader picture of the student factors that matter for learning to program, with a particular in-depth focus on affective characteristics, and also including dispositional characteristics that have not previously been studied in CS education. With a better understanding of how students experience the process of learning programming, instructional practices can be refined so that students who are interested, engaged, and able with respect to programming are not excluded simply because the learning process has a negative affective impact. This has potential application to all students, but it may particularly apply to understanding the CS gender participation gap.

CHAPTER 2

LITERATURE REVIEW

Programming is a complex skill with a difficult learning curve (Robins et al., 2003), and it can also be an emotionally challenging skill to develop (Kinnunen and Simon, 2011). These aspects of programming make it important to understand more about the learning process and the particular factors that cause difficulty for students. In order to understand how to best teach programming, It is important to understand how the characteristics and behaviors of individual students affect their experiences, and ultimately their learning outcomes. These student factors are an important area for study in CS education, because differences between student experiences arguably drive more of the variation in outcomes for students than do differences in pedagogy (e.g. Lewis and Lewis, 2008).

Detailed models based on the nature of programming tasks have been developed to explain the difficulty of learning to program. For example, the learning-edge momentum model explained the difficult learning curve of programming through the hierarchical interdependence of different programming concepts (e.g. flow of control, statements, conditions, Boolean expressions, values, operators) (Robins, 2010). Another model is the hierarchy of programming skills, which divided programming into a hierarchical ordering of skills (i.e. syntax knowledge, code tracing, code explaining, and code writing) that students' progress through as they increase in proficiency (Lopez et al., 2008).

Conversely, there is currently not a comprehensive model that explains how student factors influence learning to program. The existing research on student factors in CS education is piecemeal, with studies typically examining the influence of a small number of factors (Bergin and Reilly, 2005). From previous research in other disciplines, we know that educational outcomes depend heavily on factors related to the students as well as factors related to the material itself (See for example: Ashcraft and Krause, 2007; Champagne, 1980; Gal and Ginsburg, 1994; Vermunt, 2005), so student factors in programming demand further study. The following sections will review the existing literature on the student factors being considered in this study.

2.1 Student Factors that Influence Learning Outcomes

Previous research in CS education has examined many different cognitive and noncognitive student factors, as well as student background factors, and found many correlates of success. These cognitive factors include different mental ability constructs around problem solving, critical thinking, analytical reasoning, and mathematical ability. Non-cognitive factors include various constructs related to motivation, including self-efficacy and the use of metacognitive strategies, as well as other factors such as anxiety and comfort level. In addition to these cognitive and non-cognitive factors, factors such as previous programming experience has been empirically investigated repeatedly.

Although many individual student factors have been examined in the literature on learning to program, there remains nevertheless much that we do not yet know. One major issue with the existing literature on student factors is that the majority of the research has only examined simple correlations between single factors or small groups of factors using the easiest to measure learning outcomes, such as exam scores or overall grades. The second major issue with the existing research is that typically predictors are examined only at one time point. Despite the fact that researchers have created detailed models for the developmental process of learning to program (e.g., Lopez et al., 2008; Robins, 2010), researchers have by and large not examined student factors in the context of an ongoing learning process. It is true that some student characteristics of interest are inherently static over time, but student characteristics that are responsive to the feedback that students get from the learning process itself need to be investigated further. Such characteristics that are malleable and responsive to the learning process have the greatest potential to be influenced by changes to instructional practices, and therefore, understanding their influence could be very significant for informing improvements to instructional practices. The goal of this study was to examine these sorts of factors in more depth than they have been previously examined.

2.2 Cognitive Factors

Cognitive factors include different mental ability constructs related to problem solving, critical thinking, analytical reasoning, and mathematical ability. Prior research has examined the influence of several of these factors on student learning outcomes in programming. For example, abstract reasoning development has been found to be a successful predictor of programming learning outcomes (Barker and Unger, 1983; Kurtz, 1980). Other cognitive skills related to problem solving, such as the ability to translate a problem statement into abstract formal notation, and the ability to choose the correct sequence of steps needed to solve a problem, have also been found to be related to student learning outcomes in programming (Hostetler, 1983). Prior research has also suggested that mathematical ability is a good predictor of success in learning to program (Bennedsen and Caspersen, 2005; Bergin and Reilly, 2005; Wilson and Shrock, 2001) and also that training in problem-solving strategies may contribute to success in programming courses (Evans, 1988).

2.2.1 Problem Solving

Among the cognitive factors, problem solving has been of significant interest to computer science education researchers. Specifically, there has been a long-standing interest in programming as a means for improving students' problem-solving skills, because it is thought that problem solving closely resembles the reasoning abilities that one develops when learning to program. Problem-solving ability has been examined many times as an outcome variable in CS education research, but researchers have not consistently found evidence that learning programming bolsters generic problem-solving ability (Palumbo, 1990).

A few studies have examined problem solving as a predictor of programming learning outcomes. Findings from these studies have suggested that generic problem-solving ability may be associated with course outcomes in programming (Evans, 1988; Hostetler, 1983; Mccoy and Orey, 1988; Nowaczyk, 1984). For example, Nowaczyk (1984) examined how students' initial problem-solving skills influenced course outcomes in introductory programming. The author measured students' generic problem-solving ability using seven exercises focused on logical operations, algebraic solutions, transformations, or identification of mathematical relationships. The authors found that individual differences in problem-solving ability were significantly related to programming learning outcomes.

Prior research on problem-solving ability and student success in an introductory programming course has found that problem solving predicted course outcomes, but only for project score outcomes, and not multiple-choice exam score outcomes (Lishinski et al., 2016a). Another study examined whether the relationship between problem solving and learning outcomes in programming differed by gender. The results of this study indicated that problem-solving ability predicted project score outcomes, but only for male students (Lishinski et al., 2017a). These prior studies utilized problem-solving items from the PISA assessment to measure students' problem-solving abilities as these items are based on strong theoretical foundations and prior work by Mayer (1992) on problem solving.

2.2.2 Metacognitive self-regulation

In this study, metacognitive self-regulation was considered as one of the cognitive factors given that it consists of the self-monitoring and control strategies used by students to select and implement appropriate behaviors to address an academic challenge (Corno, 1986). Examples of such strategies include asking oneself questions to check comprehension and setting specific goals when studying. Metacognitive strategies have been found empirically to be an important predictor of academic success (Corno, 1986; Broadbent and Poon, 2015), because they mediate between cognitive strategies and academic performance (Winne, 1996). Whereas cognitive strategies are employed to complete a specific task, metacognitive strategies are more general strategies that involve awareness and executive control of the cognitive strategies. Previous research has found that students might fail to successfully perform a task because they fail to appropriately use metacognitive strategies, despite using appropriate cognitive strategies (Corno, 1986; Ford et al., 1998). Metacognitive self-regulation is a component of self-regulated learning (SRL), which is discussed further in section 2.3.

Prior work has examined metacognitive self-regulation in CS and results have suggested a positive association with learning outcomes. For example, Bergin et al. (2005) found that use of metacognitive strategies is positively associated with course learning outcomes in CS1. A qualitative study of students in an object-oriented programming course found that failures of metacognitive control accounted for problem-solving failures (Havenga, 2015). Another qualitative study detailed the types of metacognitive strategies used by introductory programming students, including assessment of task difficulty, problem decomposition, and time-management and planning (Falkner et al., 2014).

2.3 Motivational and Affective Factors

Affective factors have not been studied in great depth with respect to programming. While a number of these individual factors have been measured and studied in the CS context, many more such factors need to be investigated, and in greater depth, by studying multiple constructs simultaneously and using repeated measures. Affective factors are known to have a significant impact on the learning process in other disciplines, thus they should be investigated further in CS as well.

2.3.1 Motivational Factors

This section gives an overview of the motivational components of self-regulated learning (SRL) that are investigated in this study. Self-regulated learning refers to one's ability to understand and control one's learning environment and includes affective (e.g., motivational states, self-efficacy) and behavioral (e.g., goal setting) characteristics. A self-regulated student is someone who is behaviorally, metacognitively, and motivationally an active participant in their own learning (Zimmerman, 1986, 1990).

2.3.2 Self-efficacy

Self-efficacy is an important SRL construct that has its origins in Bandura's social-cognitive theory (Bandura, 1977, 1986). Bandura (1977) defined self-efficacy as the belief that one can successfully execute the behaviors needed to produce a desired outcome. Self-efficacy beliefs influence the amount of effort people are willing to expend and how well they cope and persist in the face of challenges (Bandura, 1977). Bandura (1986) also argued that self-efficacy beliefs often explain weak linkages between prior ability and achievement. Since the original presentation by Bandura (1977), self-efficacy has emerged as one of the most important motivational variables for explaining the relationship between past performance and future learning outcomes in education research (Valentine et al., 2004; Andrew, 1998; Bandura, 1995, 1997; Britner and Pajares, 2006; Locke and Latham, 1990; Maddux, 1995; Pajares and Miller, 1994; Pajares and Valiante, 1997; Pintrich and Schunk, 1989; Multon et al., 1991; Prat-Sala and Redford, 2012; Lee et al., 2014; Gore, 2006; Pajares and Valiante, 2012; Kingston and Lyddy, 2013).

The importance of self-efficacy in predicting academic success and choices (e.g. choice of major) has been well documented across a number of subject areas, such as mathematics (Pajares and Miller, 1994), science (Britner and Pajares, 2006), and language arts (Pajares and Valiante, 1997). For example, students' sense of self-efficacy in mathematics has been found to be more closely related to their persistence in mathematics than their previous math achievement (Pajares and Miller, 1994). Additionally, subject specific self-efficacy has also been found to significantly predict choice of mathematics and science careers (Lent and Hackett, 1994). Valentine et al. (2004) conducted a meta-analysis on the effects of self-efficacy on achievement outcomes across subject areas, and found that, after controlling for prior achievement, self-efficacy had a significant effect on student achievement. Prior research has also suggested that higher self-efficacy influences performance by initiating more adaptive behaviors, causing students to pursue more opportunities for practice and feedback (Pintrich and De Groot, 1990). Along these lines, self-efficacy has been shown to be related to more resiliency in response to complex and difficult tasks (Schunk, 1995).

Not only does students' self-efficacy influence their academic outcomes, but prior work has also suggested that self-efficacy beliefs are mutable based on students' experiences with an activity. This suggests that teachers can plausibly strengthen or manipulate students' self-efficacy beliefs as they engage in particular learning activities (Margolis and McCabe, 2006). Self-efficacy beliefs are particularly mutable early in the process of learning something new, and become intransigent as more experiences are accumulated (Bandura, 1977). Empirical studies in a variety of fields, such as Medicine (French et al., 2014), have also found that self-efficacy beliefs are subject to change in response to many contextual factors, and furthermore, increases in self-efficacy are associated with improvements in outcomes (Gecas, 1989; Gist, 1992). The combination of the known influence of self-efficacy on outcomes and its malleability makes self-efficacy a promising avenue for pedagogical interventions.

Research on self-efficacy has also examined its possible role in a reciprocal feedback loop with learning outcomes. In this context a reciprocal feedback loop means that self-efficacy influences later performance, which in turn influences later self-efficacy, and so on. In Bandura's initial formulation of self-efficacy, he suggested that self-efficacy and performance should be examined at significant junctures in the learning process because they are expected to reciprocally affect one another (Bandura, 1977). Empirical studies attempting to verify this reciprocal relationship are few, but one large scale (multinational) study using PISA data has found evidence that the reciprocal feedback loop relationship exists (Williams and Williams, 2010). It is important to prevent students from experiencing a negative selfefficacy feedback loop; previous research on self-efficacy has suggested that one thing that may stabilize student self-efficacy and prevent it from dropping in response to negative performance feedback is giving students an opportunity for self-evaluation (Schunk and Ertmer, 2000). The idea is that by getting students to self-evaluate how and why their performance was not satisfactory, they are required to use self-monitoring and reflection strategies that are known to be correlated with academic success.

2.3.3 Goal orientation

Another important component of self-regulated learning theory is goal orientation, which refers to the types of outcomes students desire for academic achievement situations. Goal orientation is a significant component of SRL because it defines the standards that students use when monitoring their learning process (Locke and Latham, 2012). Students' goal orientations fall under two main categories - mastery and performance. Students with a mastery goal orientation (also known as intrinsic goal orientation) value learning and personal growth, whereas students with a performance goal orientation (also known as extrinsic goal orientation) value social demonstration of success and relative achievement (Pintrich and Schunk, 1995). Prior research has suggested that having an intrinsic goal orientation is more conducive to academic success whereas having an extrinsic goal orientation often leads to academic difficulties (Wolters and Yu, 1996).

2.3.4 Relationships between motivational constructs

Previous research has also shed light on the relationships between these different selfregulated learning constructs. For example, self-efficacy has been found to be positively associated with the use of metacognitive strategies (Pintrich and De Groot, 1990). Similarly, intrinsic goal orientations have been shown to be positively associated with self-efficacy, whereas extrinsic goal orientations are negatively associated with self-efficacy (Wolters and Yu, 1996). Metacognitive self-regulation has also been found to be significantly related to later self-efficacy, possibly mediated by the effects of performance on self-efficacy (Al-Harthy and Was, 2013). When considering all three constructs, a meta-analysis by Credé and Phillips (2011) found that self-efficacy consistently had the strongest relationship with academic outcomes, followed by metacognitive self-regulation and then goal orientation. These previous results provided the basis for the model of the relationships between these constructs used in this study.

2.3.5 Motivational Factors in CS education

Previous research in computer science education has examined the individual relationships between motivational constructs (self-efficacy and goal orientations) and student learning outcomes in CS courses. Self-efficacy has been the most studied, with previous research mostly confirming that self-efficacy predicts performance in CS contexts. For example, Ramalingam et al. (2004) examined novice programming students and found that their self-efficacy was positively associated with course grades and previous experience, and that it increased over time during the course. In another study, Wiedenbeck (2005) investigated the influence of self-efficacy on novice programmers' success at a debugging task using a computer programming self-efficacy scale. The results suggested that self-efficacy was positively associated with both overall course grade as well as outcomes on the debugging task. Watson et al. (2014) found that self-efficacy was the strongest predictor of performance in an introductory programming course (commonly known as CS1), above many other predictors of success, including traditional predictors like previous programming experience as well as data mining based aggregates of observed programming behaviors. Wilson and Shrock (2001), however, found that self-efficacy was not significantly related to course midterm grades.

A qualitative study by Kinnunen and Simon (2011) found that CS1 students' selfefficacy judgments were continually evolving during the process of working on programming assignments. Both positive and negative self-efficacy judgments were possible responses to both positive and negative learning outcomes, and task difficulty and achievement goal orientation were possible moderators of these changes (Kinnunen and Simon, 2011). A precursor to this study found that programming assignments tend to surprise students with problems, even when they think that they understand what they are doing (Kinnunen and Simon, 2010). These "struck by lightning" experiences produce particularly intense emotional reactions in students which have accordingly larger effects on students' revisions of their self-efficacy beliefs (Kinnunen and Simon, 2010).

The research on goal orientations in CS courses has found positive correlations between intrinsic goal orientations and course grades. For example, Bergin et al. (2005) examined the relationship between goal orientations and outcomes in a CS1 course, and found that intrinsic goal orientations were associated with higher exam scores in the course. Similarly, Zingaro and Porter (2016) found a connection between intrinsic goal orientation and course outcomes and interest in CS with CS1 students.

Previous work has also suggested that there are reciprocal relationships between self-efficacy and learning outcomes in programming (Lishinski et al., 2016b). Using path analysis, evidence of reciprocal relationships between self-efficacy and course outcomes was found. This study also found evidence of gender differences in the reciprocal self-efficacy process, with the data suggesting that female students enacted the reciprocal process of revising self-beliefs in accordance with feedback significantly quicker than did male students (Lishinski et al., 2016b).

2.3.6 Emotional Responses

Students' emotions while doing academic tasks have been studied in a variety of settings, and have been shown to be related to affective characteristics and academic outcomes (Linnenbrink-Garcia and Pekrun, 2011). One study found that emotions experienced while doing homework assignments were different from emotions experienced in the classroom, and that homework emotions were linked to academic self-concept and achievement outcomes (Goetz et al., 2012). Another study found that unpleasant emotions during homework sessions were negatively related to both effort and achievement, and that unpleasant emotions were driven by the perceived quality of the homework assignment (Dettmers et al., 2011).

Of particular relevance to the programming context, previous research has suggested that emotions students experience during problem solving significantly influence their self-regulation (Hannula, 2015). Prior research has also examined the role of students' emotional experiences while learning to program. Eckerdal et al. (2007) investigated students' emotional reactions when learning threshold concepts. Threshold concepts can be seen as the core concepts within a discipline around which curriculum should be organized (Mead et al., 2006). Through interviews, Eckerdal et al. (2007) learned that students often felt frustrated, depressed, and humbled, but also that they eventually became confident when they had succeeded in grasping the construct.

Kinnunen and Simon (2010) offered a detailed qualitative account of the emotional experiences that novice programming students have while working on programming assignments. Kinnunen and Simon (2010) examined students' emotional responses to programming assignments using open-ended qualitative interviews. These initial interviews elicited high level reflections from students at the end of the first year of starting a programming course of study. It is important to note that they did not have access in this study to students' immediate emotional responses to programming projects, because they interviewed students later. They summarized students' emotional responses with five categories: Proud/Accomplished, Frustrated/Annoyed, Inadequate/Disappointed, Relaxed/Relief, and Apprehensive/Reluctant.

Kinnunen and Simon (2011) continued this research with a study that examined how students' emotional responses influenced their self-efficacy beliefs. The results suggested that both positive and negative programming experiences could result in either positive or negative self-efficacy assessments. The combination of positive experiences and positive selfefficacy judgment, as well as negative experiences with negative self-efficacy judgments are unsurprising, whereas the other two combinations merit further explanation. These combinations mean that students make counterintuitive self-efficacy assessments in some situations. For example, the authors found that some students who had a negative programming experience could maintain a positive self-efficacy judgment, being confident that they would eventually get it, despite current difficulties. By contrast, the authors found that other students who have had a positive programming experience, such as finishing the project, may have also received other cues that lead them to maintain a negative self-efficacy assessment. In order to help students avoid difficulties, the authors recommended that programming instructors avoid inducing repeated failures early on in the learning experience, that they set concrete attainable subgoals for projects, that they should assign tasks at an optimal difficulty level which slightly extends the student's current state, and that they should foster classroom environments that encourage a mastery orientation.

Previous investigations of students emotional responses to programming assignments have shown that there is a significant connection between early emotional responses and later course outcomes, as well as evidence of reciprocal relationships between emotional responses and outcomes. Students' feelings of frustration and confidence that they will do well after early projects were found to be significantly correlated with project performance at the end of the course (Lishinski et al., 2017b). Path analysis results also showed significant indirect effects, providing evidence that a feedback loop process can occur with emotional responses and later project performance (Lishinski et al., 2017b). Given the salience of emotional episodes particular to programming, as documented by Kinnunen and Simon and by these previous results, as well as the significant impact of emotions found in previous educational psychology research, the topic of students' emotional reactions to programming projects merits further investigation.

2.3.7 CS gender participation gap and the role of motivational factors

It is well known that there exists a gender gap in computer science (CS) participation, with women pursuing CS degrees at a far lower rate than men. Women hold 48% of all jobs in the US, but earn just 22% of all undergraduate CS degrees. Despite the attention paid to the gender gap, we still know relatively little about factors that influence the lack of female participation in CS. The participation gap may be explainable in part by difference in students' affective experiences, as prior research has found systematic differences between male and female students' experiences in programming courses. For example, female students have been observed to have lower self-efficacy and comfort levels than their male peers (Beyer, 2008; Busch, 1995). Other studies have found that attributions of success in the CS context differ between genders, with female students being more likely than male students to attribute success to luck rather than ability (Cohoon, 2003; Wilson, 2010). Prior research has also suggested that women's performance in CS courses is significantly related to comfort level and perceptions of gender equity, which is not the case for male students (Bernstein, 1991; Beyer, 2008). These results indicate the importance of affective experiences and characteristics, which also underscores the importance of further investigation of these factors, in the context of the gender gap in CS education.

2.4 Dispositional Factors

Personality traits are constructs that serve to label particular enduring behavior patterns, which serve to characterize both individuals and differences between individuals (Mischel, 2013). Personality traits have long been recognized as an important contributor to academic achievement (Komarraju et al., 2009), and have been shown to account for variation in academic outcomes above and beyond intelligence (Kappe and Van Der Flier, 2012). Researchers have argued that personality traits are a particularly good predictor of postsecondary education outcomes (Connor and Paunonen, 2007). Models of personality based on discrete factors (traits) goes back to work by Cattell in the 1940s, but the particular Big Five model of personality discussed here originates from work by Goldberg in the early 1990s. Goldberg (1990) proposed a model of personality based on five orthogonal factors. The five personality factors that came out of this work were labelled Surgency (also labelled as extraversion in subsequent work), Agreeableness, Conscientiousness, Emotional Stability (also labeled as neuroticism in subsequent work), and Intellect (also labelled openness to experience in subsequent work). The Extraversion/Surgency factor refers to tendencies to be sociable and seek stimulation through social interaction. The Agreeableness factor refers to tendencies to be trusting, to cooperate, and to help others. The Conscientiousness factor refers to tendencies to be organized, disciplined, and dependable. The Neuroticism/Emotional Stability factor refers to tendencies to experience negative emotions such as anxiety and depression. The Intellect/Openness to Experience factor refers to tendencies to be intellectually curious and willing to try new things.

The personality traits in the big five model have been found to be strongly related to academic outcomes for postsecondary students (Connor and Paunonen, 2007; Kappe and Van Der Flier, 2012; Komarraju et al., 2009). Previous findings have suggested that conscientiousness can explain five times more variance in college GPA than intelligence (Kappe and Van Der Flier, 2012). Some of the research on the big five has suggested that the relationship between the big five and academic outcomes has to do with influences of personality traits on intrinsic motivation. For example, conscientiousness has been found to be highly correlated with intrinsic motivation (Kappe and Van Der Flier, 2012), whereas other studies have suggested that conscientiousness is a mediator between intrinsic motivation and student success outcomes (Komarraju et al., 2009). Research on the big five personality traits in computer science education has been rare. One study examining pair programming found that the big five personality traits had no impact on the effectiveness of pair programming in increasing student learning outcomes (Salleh et al., 2009).

2.5 Models of Cognitive Skill Development

Programming is a complex cognitive skill, which involves several important cognitive components, such as mental representations of programming constructs, working memory, field dependency, and problem decomposition (Brooks, 1999; Mancy and Reid, 2004; Mayer et al., 1986). There is a related body of research in applied psychology on skill acquisition, which covers a broad range of settings where the central focus is on teaching people how to do something. This research is applicable to situations in education, like programming courses, where the end goal is a skill. The instruction that students get in an introductory programming course is not focused on the mastery of a fixed set of factual materials. While there are facts about programming that should be learned, introductory programming courses do not typically assess them as a learning outcome. In the setting for this study, students are never tested on factual material in the course. The focus of the course is to inculcate the students with a level of mastery of programming. Thus, we can approach learning to program as an instance of skill acquisition, and look to research on this topic for guidance.

Research on complex cognitive skill acquisition has sometimes involved models in which the interaction of discrete cognitive component skills is used to explain student learning on complex tasks (Ackerman, 1988). However, research on complex skill acquisition has also shown non-cognitive factors to be just as influential on student outcomes as cognitive factors in generic skill acquisition contexts (Kanfer and Ackerman, 1989). To that end, Kanfer and Ackerman (1989) built a model for understanding complex skill acquisition based on the interaction of cognitive and motivational factors. Their specific model was built on the interactions between limited attentional resource capacity, dynamic resource demands of tasks, and the influence of motivation on task effort, but the core idea is using a combination of the affective and cognitive demands of a task to explain learning outcomes. Therefore, given that programming is a complex cognitive skill, a model of the factors that influence the development of programming skills should incorporate cognitive as well as affective factors, but even more importantly, their interaction with one another.

2.6 Research Questions

This study investigated students' cognitive, affective, and dispositional characteristics in order to address these four research questions:

- RQ1: How do cognitive, affective, dispositional factors interact and influence learning outcomes for beginning programming students?
- RQ2: What are the most important factors that influence student learning outcomes in programming?

- RQ3: How do students' emotional responses to programming projects influence their learning outcomes over time?
- RQ4: How effective are interventions targeting self-efficacy, and how do they influence student outcomes?

As detailed throughout this section, many studies have investigated cognitive determinants of learning outcomes in educational psychology, as well as in CS education. Likewise, there are an abundance of studies investigating the affective and dispositional determinants of learning outcomes. Given the documented importance of these student factors, the purpose of this study was to develop a detailed model of cognitive, affective, and dispositional factors that influence learning outcomes in introductory programming. Combining these factors into a single model that reflected their complex interrelationships was the purpose of research question 1. This study also sought to determine which of all of the factors were the strongest predictors of learning outcomes, which is addressed by research question 2. Students' emotional responses, being so specific to the experience they were associated with, called for a different sort of analysis looking at the effects of these emotional responses across many points in time. For that reason, the emotional response data was analyzed separately under research question 3.

This study also sought to investigate whether an intervention could positively influence students' self-efficacy and course outcomes. Self-efficacy was singled out as the subject of further investigation for several reasons. First, its strong influence on academic outcomes has been seen repeatedly in previous research, both within CS education, and in educational psychology generally. Second, it is perhaps the most malleable of the student affective characteristics being examined in this study. Third, of the factors examined in this study, it has received the most research attention in CS education. This previous research in CS education has established its predictive value for CS learning outcomes, as well as alerted CS education researchers to its importance. For both of these reasons, it was important to go beyond merely attempting to link self-efficacy to outcomes, and to investigate whether this factor that is already known to be important can be influenced in order to create better learning outcomes for students. This goal is addressed by research question 4.

CHAPTER 3

METHODOLOGY

3.1 Design of the Study

The main goal of this study was to develop a model of how cognitive, affective, and dispositional factors interact to influence student learning outcomes in programming. To create this model, data was collected from a large cohort of introductory programming students on cognitive, affective, and dispositional characteristics. The cognitive constructs of interest included problem solving and metacognitive self-regulation; affective constructs included self-efficacy, goal orientation, and emotional responses to programming assignments; and dispositional constructs included the big five personality traits (conscientiousness, agreeableness, emotional stability, openness to experience, and extraversion). This study focused on the exploratory modeling of important predictors of learning outcomes in order to provide a better big picture account of the important student factors.

This study also included an intervention designed to bolster students' self-efficacy. The intervention involved asking students to complete a self-evaluation task in order to reduce the possibility of a negative self-efficacy feedback loop (Lishinski et al., 2016b). The hypothesis was that the self-evaluation task would force students to engage in some metacognitive reflection on their performance, which has been shown to increase self-efficacy and in turn learning outcomes (Schunk and Ertmer, 2000). If such changes in self-efficacy were observed associated with the intervention, a corresponding increase in performance on subsequent assignments would be expected.

3.2 Sample of Study

The population of this study was undergraduate students enrolled in an introductory programming course (CS1) at a large Midwestern university. Participants in this study consisted of 612 CS1 students, who were recruited to complete the measures used in this study. The participants were mostly male (N=481; 78.6%), but included 131 female students (21.4%). The breakdown of race/ethnicity for the participants is shown in table 3.1. The students in this study came from a wide variety of majors, the top 20 of which are shown in table 3.2 (note that Computer Science and Mechanical Engineering students were required to take this course).

Table 3.1: Race/Ethnicity Breakdown

Race/Ethnicity	Percentage of students
White (non-Hispanic)	74.8
Asian (non-Hispanic)	14.5
Black or African American (non-Hispanic)	5.1
Hispanic Ethnicity	3.0
Two or more races (non-Hispanic)	2.1
American Indian/Alaskan Native (non-Hispanic)	0.4

3.3 Measures

3.3.1 Problem Solving

A problem-solving instrument, derived from the 2003 PISA (Program for International Student Assessment) problem-solving assessment, was used to measure one of the cognitive factors: students' problem-solving abilities. PISA defined problem solving as: "An individual's capacity to engage in cognitive processing to understand and resolve problem situations where a method of solution is not immediately obvious" (PISA, 2003). The PISA problemsolving items required students to reason through novel and realistic problems, which could not be solved by merely using heuristics from previously acquired content knowledge. The PISA problem-solving items were divided into three categories: decision making, systems analysis and design, and troubleshooting. Decision making items required students to come to an optimal decision about a course of action given pre-specified constraints of a system. Systems analysis and design items required students to use information about how a sys-

Major	Percentage of students
Computer Science	28.1
Mechanical Engineering	25.2
Computer Engineering	8.0
Mathematics	4.1
Media and Information	2.9
Economics	2.8
Statistics	2.6
Actuarial Science	2.1
Electrical Engineering	2.0
Supply Chain Management	1.8
Finance	1.6
Accounting	1.5
Business-Admitted	1.1
Experience Architecture	1.1
Physics	1.1
Applied Engineering Sciences Human Biology Chemical Engineering Biochem&Molecular Biol/Biotech Business-Preference	$ \begin{array}{c} 1.0 \\ 1.0 \\ 0.8 \\ 0.7 \\ 0.7 \end{array} $

Table 3.2: Top 20 Majors

tem was structured and constrained to reason about the effects of changes to the system or external events. Troubleshooting items required students to reason about possible causes for events that are unexpected given the purpose and design of a system. PISA released ten problem-solving items after the test, and each item consisted of a situation description, along with 1-3 questions of varying difficulty.

This study used four adapted items from the problem-solving section of the 2003 PISA assessment to measure students' problem-solving ability. The four items used in this study were the library system item, the trip item, the irrigation item, and the design by numbers item, which together had 10 associated questions (see Appendix D). These items were selected because they were reported by PISA as empirically the most difficult items (PISA, 2003), they had the most associated questions (each item had between one and three associated questions), and the type of logical deductive reasoning required to solve these problems seemed the most relevant to programming students (Lishinski et al., 2016a). The items were adapted for use on an online survey platform (qualtrics) by replacing hard to implement elements with easier to implement versions. For example, the original library system item required students to draw a flowchart by hand, whereas the implementation of this question on qualtrics required students to fill in the blanks of a flow chart.

Problem solving ability was operationalized as students' overall score on the four problem-solving items. Overall scores were weighted towards the more difficult questions (based on the PISA reported percentage correct) to maximize variability. Furthermore, the scoring system was adjusted to allow for a more fine-grained assessment of problem-solving capabilities. In the original test, items were scored as correct/incorrect/partial credit. In this study, by contrast, each individual component of questions requiring multipart responses were scored separately. These decisions were made because the items were administered under strict time constraints. For this study, students were given just 15 minutes to complete all four items, to ensure that they were appropriately difficult for the undergraduate population of students being studied. Prior results from studying this student population have shown that the desired difficulty was achieved with roughly normal score distributions and no evidence of either floor or ceiling effects (Lishinski et al., 2016b).

The four items were administered via qualtrics in random order with enforced time limits for each item. The original free-form items were converted into multiple blank short answer questions for delivery via qualtrics and easier scoring. The items were scored using an automated rubric program, which assessed whether student responses matched either the correct answer or a small range of possible correct answers, which were included with the released items. The problem-solving survey was administered twice in the semester, near the beginning and near the end.
3.3.2 Motivated Strategies for Learning Scales (MSLQ)

In order to measure students' metacognitive self-regulation, as well as the motivational constructs self-efficacy, and intrinsic and extrinsic goal orientations, four subscales from the Motivated Strategies for Learning Questionnaire (MSLQ) were used. The MSLQ includes 83 items across 15 subscales, which cover constructs from two main areas: motivation and learning strategies (Pintrich et al., 1993). In order to measure the constructs of interest for this study, four scales were chosen, which consisted of 28 seven point Likert items (see Appendix A): self-efficacy, intrinsic goal orientation, extrinsic goal orientation, and metacognitive self-regulation. These scales were chosen due to the theoretical importance of these constructs in self-regulated learning, as well as the previously documented reliability of the subscales. Furthermore, the prior use of these specific constructs in CS education research has suggested their importance in predicting outcomes for CS students (see section 2.3.5).

The metacognitive self-regulation subscale of the MSLQ consists of 12 Likert items that ask students to assess the extent to which they engage in various self-regulatory behaviors. The self-efficacy subscale of the MSLQ consists of eight Likert items that ask students to assess their level of confidence in their ability to successfully master the material in the course. The intrinsic goal orientation subscale of the MSLQ consists of four Likert items that ask students to indicate whether their goals for the course involve mastery of the material. The extrinsic goal orientation subscale of the MSLQ consists of four Likert items that ask students to indicate whether their goals for the course involve mastery of the material.

The reliability values of the four scales used in this study have been reported by the creators of the study from a validation sample (Pintrich et al., 1993), as well as from meta-analysis results (Credé and Phillips, 2011). The previously reported coefficient alpha values for these scales, along with the observed values for the sample used in this study, are shown in table 3.3.

The reliability results for the MSLQ scales suggested that the scales reached acceptable levels of reliability with the study sample. Overall alpha values were .79 or higher

Table 3.3: MSLQ Reliability

	Previously Reported Alpha: Validation Sample	Previously Reported Alpha: Meta-analysis	Observed Alpha
Metacognitive Self-regulation	0.91	0.93	0.96
Self-efficacy	0.69	0.74	0.83
Intrinsic Goal Orientation	0.66	0.62	0.79
Extrinsic Goal Orientation	0.77	0.79	0.80

for all of the scales. Item-specific statistics did not indicate any strongly discordant items in the self-efficacy or either of the goal orientation scales. The metacognitive self-regulation scale had two discordant items, item 1 and item 8, which were the two negatively worded items (see Appendix A.1). Even after reversing their scale, the reliability analysis indicated that items 1 and 8 negatively impacted the reliability of the scale, so they were dropped. Overall, the reliability results observed with this data set compared favorably with previously reported reliability values for these scales, with the observed reliability values higher than those previously reported in every case. The four MSLQ scales were administered electronically via qualtrics and the items were presented in a randomized order to avoid item position effects. The scales were administered near the beginning of the semester concurrent with the first administration of the problem-solving items.

In order to measure students' emotional responses to programming projects, four 7-point Likert-scale items were used (see Appendix E). Three items were developed based on previous research from Kinnunen and Simon (2010), which identified five categories of emotions that students used to describe their emotions while doing programming assignments. These categories included: happy/proud, frustrated/annoyed, inadequate/disappointed, relaxed/relief, and tired/apprehensive. Only the first three of these categories were used for to create the items for this study in order to keep the survey very brief, and also because the final two categories seemed somewhat redundant and less salient (Lishinski et al., 2017b).

Each of the questions asked students to gauge their level of a particular emotion after completing a programming project. Feeling of happiness/pride: this question asked students to assess the extent to which they agreed that "upon completing the project, I felt proud/accomplished." Feeling of frustration/annoyance: this question asked students to assess the extent to which they agreed that "while working on the project, I often felt frustrated/annoyed." Feeling of inadequacy/disappointment: this question asked students to assess the extent to which they agreed that "while working on the project, I often felt inadequate/disappointed."

The fourth item on the emotional response survey was taken from the self-efficacy scale of the MSLQ, and asked students how well they thought they would do in the course. This item was chosen in order to assess students' self-efficacy judgments on a repeated basis after each of the 10 programming projects, but without asking students to respond to all eight self-efficacy items. The self-efficacy item taken from the MSLQ used the prompt "Considering the difficulty of the course, the teacher, and my skills, I think I will do well in this class."

The four 7-point Likert scale items were administered to all students 10 times over the semester by adding them to the instructions for programming projects 2-11, with students asked to insert their answer as a note at the bottom of their actual project code file. This was done in order to gauge emotional responses to projects as close to the actual experience as possible.

3.3.3 Big Five Personality Traits

The Big Five personality traits (extraversion, agreeableness, conscientiousness, emotional stability, and openness to experience) were measured using the Big-Five factor markers developed by Goldberg (1992), which included 50 Likert items for assessing individual trait values. These markers were made available through the international personality item pool (ipip.ori.org), a project developing and collecting sets of personality items that are made freely available for use by researchers (Goldberg, 1992). These items have been analyzed for validity and reliability, and the available evidence suggests that they are able to meet these requirements (Goldberg, 1999; Gow et al., 2005; Lim and Ployhart, 2006; Rammstedt et al., 2010; Zheng et al., 2008). The five personality trait scales each included ten 7-point

Likert items (see Appendix B). The previously reported reliability values alongside the values observed in this study are shown in table 3.4.

	Previously Reported Alpha	Observed Alpha
Extraversion	0.87	0.86
Agreeableness	0.82	0.82
Conscientiousness	0.79	0.79
Emotional Stability	0.86	0.85
Openness to Experience	0.84	0.81

Table 3.4: Big Five Reliability

Overall, the reliability results for the big five indicators suggested that the personality trait scales reached acceptable levels of reliability. Overall alpha values were .79 or higher for all of the scales. The item-specific statistics did not indicate any strongly discordant items in any of the scales. The reliability results observed with this data set are comparable with previously reported reliability values for these scales, with the observed reliability values being equal or only slightly lower than those previously reported.

The personality trait survey was administered via qualtrics in the second to last week of the course, concurrent with the second administration of the problem-solving items. Whereas the MSLQ scales were administered at the beginning of the class because the constructs were expected to be affected by the course itself, the personality items were administered near the end of the course for logistical reasons, and because the personality factors were not expected to change over time or as a result of the experiences in this course, because personality traits are by definition stable over time (Mischel, 2013).

3.3.4 Course Learning Outcomes

The course grade consisted of three multiple-choice examinations (two midterm and one final), which made up 45% of the course grade, 11 programming projects, which made up 45% of the course grade, and homework exercises, which made up 10% of the course grade.

The analyses for this study focused on the exam and project grades, because they made up the large majority of the course grade, and because the homework data was graded pass/fail and therefore had virtually no variation between students.

The midterm and final exams were multiple choice, and therefore limited in what they could measure of a student's understanding of programming. The questions generally focused on syntax and code tracing, which correspond to the lowest level of the hierarchy of programming skills (Lopez et al., 2008). Nevertheless, the exam data had the advantage of being collected under controlled conditions. All student responses to all individual exam questions (in addition to aggregate scores) were collected so that item response theory (IRT) analysis of this data could be conducted to explore the item response properties of different test forms and produce equivalent scores across forms (see Appendix F for an example exam).

The programming projects offered a deeper look at students' understanding of programming than the exams, because students were required to write their own code and modify code provided by the instructors. The drawback is that this data is potentially noisier than the exam data, due to the larger number of variables that go into completing a programming assignment. The projects were graded according to a rubric that varied from project to project, but with some common themes such as correct coding format, and correct implementation of subcomponents of the program (see Appendix G for an example project).

The self-evaluation intervention involved students completing a brief survey four times throughout the semester. The survey asked students to identify three things that they felt went well with their last project and three things that did not go well or that need improvement (see Appendix C). The students were encouraged to participate with a monetary reward, which they were given for completing the survey four times, after projects 4,5,6, and 7. These projects were selected because they were in the middle of the course, which provided data before the intervention as a control and data after the intervention as a way to examine whether any observed effects endured after students stopped doing the self-evaluation task. The opportunity to complete the survey was offered only to a randomly



Figure 3.1: Data Collection Timeline

selected group of half the students in the class, with the other half of the class serving as a control group.

3.3.5 Data Collection

The data collected for this study consisted of the five surveys detailed above, as well as the course exam and project scores. The timeline of data collection is detailed in figure 3.1.

3.4 Data Analysis

This section describes the analyses that were conducted to answer the research questions for this study. The first subsection describes the use of Item Response Theory (IRT) models to provide a measurement model for both the problem-solving items and the exam score data. The following subsections describe the specific analyses that were used to answer each of the four primary research questions in this study.

3.4.1 IRT

Item response theory (IRT) allows researchers to apply statistical models to better understand the properties of items from tests and survey instruments, grounded in the notion that observed responses to items on an instrument are a function of some underlying latent trait (Fan and Sun, 2013). The primary IRT application of interest for this study is latent trait scoring. Previous research has shown that spurious findings can result from improper use of raw scores, and that IRT-based latent trait scoring approaches can ameliorate these potential issues (Reise et al., 2005; Kang and Waller, 2005). For this study, IRT models were applied to the data from the problem-solving assessment data and the multiple-choice exam data to estimate latent trait scores.

IRT models were used in this study for a couple of different reasons. In the case of the multiple-choice exams, the use of IRT allowed equivalent overall scores across different exam forms to be calculated. In the case of the problem-solving items, the use of a bifactor model allowed the correlation between questions within each item to appropriately be accounted for. Also, because the problem-solving items were being scored differently than they were in the PISA test, it was important to have a measurement model for this data. IRT models were particularly beneficial for the problem-solving score estimation because it enabled the use of previously collected data to improve the individual score estimates. The problem-solving measure had been used in several previous semesters of data collection, and there were over 4000 previous observations that were used to fit the measurement model. These previous observations were a valuable source of information on the properties of the different items being used, and the use of IRT models allowed for the use of this information to make more accurate estimates of individual students' problem-solving ability.

The course included three multiple-choice exams given across the semester. The exams were done on paper and the questions asked students to mentally evaluate a small snippet of code and determine what its output would be. The exams consisted of 25, 30, and 40 questions respectively, and virtually all of the questions were multiple choice with five answer choices. Furthermore, for each exam three distinct forms were used, divided among students randomly. For the purposes of this study, the exam scores were considered to represent a broad, univocal construct regarding students' understanding of the syntax and semantics of Python. In order to best achieve this measurement goal for the exam scores, and to ensure that exam scores between different forms of each exam were construct equivalent, a unidimensional IRT model was fit to the exam data to produce EAP scores for

each student. EAP scoring is a Bayesian approach to scoring in which a standard normal prior distribution for latent trait values is combined with the likelihood function for each observed item response to create the posterior distribution of latent trait estimates. The mean (expectation) of this posterior distribution provides the score estimate. The 2PL item response model was fit to this data with a fixed pseudo guessing parameter. The 2PL model includes two item parameters — item difficulty and item discrimination. Because the items were five-option multiple choice, the pseudo guessing parameter was fixed at 0.2. The models were fit using the mirt package version 1.20.1 (Chalmers, 2012) in R version 3.4.0 (R Core Team, 2017).

The problem-solving data consisted of 10 questions that correspond to four problem situations. Furthermore, because most of the questions consisted of more than one "blank", the items were scored such that the student data consisted of 31 dichotomously scored blanks. Problem solving was considered for the purposes of this study to be a univocal construct which is assessed by the each of the questions. Nevertheless, because the questions were grouped (with respect to both content and presentation) within four distinct problem situations, and understanding the situation itself is necessary for success on all of the items within each group, it would be expected that there is local dependence between items within each problem situation. These groups of dependent items are known as testlets, and their presence makes standard IRT models, with their assumptions of unidimensionality, inappropriate (Li et al., 2006). The IRT model that is appropriate for use with testlets is known as the bi-factor model (Gibbons and Hedeker, 1992). The bi-factor model consists of one primary factor, on which all the items load, as well as multiple secondary factors for each of the testlets, on which the items in each testlet load (Li et al., 2006). The bi-factor model was fit to the problem-solving data using one primary factor and four secondary factors corresponding to the four problem situations. Maximum Aposteriori (MAP) scores were calculated for each individual after the model was fit, because this scoring method is recommended over EAP for multidimensional models with more than three dimensions (Chalmer



Figure 3.2: Problem Solving Sum vs MAP scores

et al., 2017). A scatter plot comparing problem-solving sum scores to MAP scores is shown in figure 3.2.

3.4.2 RQ1 Data Analysis

To determine how cognitive, affective, dispositional factors interact and influence learning outcomes, path analysis was used to test the hypothesized patterns of relationships between the many individual variables. Path analysis is a structural equation modeling technique that allows one to model complex, interrelated patterns of explanatory relationships between many observed variables (Raykov and Marcoulides, 2006). Path analysis can be considered to be an extension of multiple regression, in that the path model specified by the researcher contains multiple sets of relationships that would individually specify a single multiple regression model (Garson, 2008). With path analysis, the researcher can specify some independent variables to explain any of the variables in the model, which can then in turn explain further variables, whereas multiple regression allows for only one dependent variable explained by all of the independent variables. This makes path analysis well-suited to examine the complex set of interrelationships hypothesized between the variables being examined in this study.

The path analysis began with an initial path model containing all of the hypoth-

esized relationships. The initial path model is shown in figure 3.3; the same initial model was used for both the project and exam score outcomes. The initial path model was derived from two main sources. First, it was based on prior data collection in this introductory CS course (See Lishinski et al., 2016b). Based on the results of previous path analyses conducted with that previously collected data, some paths were hypothesized to be significant in this study. For example, results from a previous study (Lishinski et al., 2016b) showed that the path from metacognitive self-regulation to self-efficacy was significant, so it was included in the initial model. Second, it was based on other previous research on these constructs in educational psychology, or in the context of other disciplines like math or science (Bell and Kozlowski, 2002; Bidjerano and Dai, 2007; Caprara et al., 2011; Chen et al., 2004; Credé and Phillips, 2011; Judge et al., 2002; Karwowski et al., 2013; Pajares and Miller, 1994; Pajares, 1996; Thoms et al., 1996; Vermetten et al., 2001). These results provided other hypothesized paths that were incorporated into this initial model. For example, a study by Vermetten et al. (2001) found a significant connection between openness to experience and intrinsic goal orientation, so this was included as a path in the initial model. Furthermore, the initial model was also constructed in accordance with causally reasonable restrictions on the theoretical connections between constructs (e.g. more fixed characteristics like personality traits affect more fluid characteristics like self-efficacy rather than vice versa).

The procedure for the path analysis (conducted separately for the project scores and exam scores model) went as follows. First the initial model was fit to the data. Then the model was assessed for overall fit using the χ^2 and RMSEA. The χ^2 test statistic (minimum function test statistic) is a measure of the degree to which the observed covariance matrix differs from the model-implied covariance matrix. The p-value of this test statistic therefore indicates the significance of the difference between these covariance matrices, so a p-value below 0.05 indicates a significant lack of fit and p values above 0.05 do not enable one to infer that there is a significant lack of fit. Therefore, an acceptable model fit requires a χ^2 p-value greater than 0.05. The root mean square error of approximation (RMSEA) is an



Figure 3.3: Initial Path Model

alternative fit index for structural equation models, that ranges between 0 and 1, where a value less than 0.05 is considered to indicate that the model is a reasonable approximation of the data (Browne and Cudeck, 1993). The 90% confidence interval for the RMSEA value is also frequently examined alongside the point estimate, and if the left endpoint of this confidence interval is considerably smaller than 0.05 and the interval sufficiently narrow, this can be used to infer that the model is a plausible means for describing the data (Raykov and Marcoulides, 2006). Both fit indices are included because the χ^2 test statistic is sensitive to sample size whereas RMSEA is not (Raykov and Marcoulides, 2006).

If the fit of the model was inadequate, the modification indices for the model were examined. Modification indices provided information about what paths or covariances could be added to the initial model to improve the fit. Therefore, by adding parameters to the model that have a high modification index, the fit of the model was able to be improved. For this study, the modification indices were used to select parameters to add to the initial model in order to improve the fit, and once the model fit was improved to acceptable levels by this process, the resulting model, called the modified model, was interpreted to determine which variables had the greatest impact on the project and exam score outcomes. The standardized path coefficients are presented in the model parameter tables, and in the path diagrams; the path diagrams include only the significant standardized path coefficients. Standardized path coefficients use standard deviation units, so that coefficients are all directly comparable with one another. The interpretation of the standardized path coefficients is done in standard deviation units as well, so for example, a coefficient of 0.2 means that a 1 standard deviation increase in the independent variable is associated with a 0.2 standard deviation increase in the dependent variable.

3.4.3 RQ2 Data Analysis

Multiple regression analysis was used to investigate the connections between student learning outcomes and the cognitive, affective, and dispositional characteristics of interest (initial problem-solving scores, personality traits, and SRL constructs). The same model was used with both total exam scores and total project scores as the outcome (dependent) variable in order to determine which factors explained the most variance for the different learning outcomes. Multiple regression was suitable for determining which of the factors being examined were the most important given all the others, but without having to account for possible causal relationships between the variables. Therefore, the multiple regression model gave a more straightforward overview of the most significant student factors, whereas the path analysis in the previous section was able to present a more nuanced picture of the way the factors interact with one another.

The difference between path analysis and multiple regression is important for identifying the largest predictors of the outcome variables. Multiple regression identifies which of the independent variables are the biggest predictors of the outcome variable, because the model includes the relationship between all of the independent variables and the outcome variable. Path analysis models, on the other hand, only include the relationships between the independent variables and the outcome variable that are specified by the researcher. If the strongest predictor is not posited in the path analysis model to be related to the outcome variable, the model will not identify the overall strongest predictor of the outcome variable. Multiple regression was used to ensure that, in addition to testing the relationships posited in the path analysis model, the overall largest predictors were also identified.

The multiple regression analysis was conducted in a hierarchical multiple regression framework, in which blocks of predictors were entered into the model separately and successively to further examine the effects of the groups of variables on the outcomes. The independent variables were separated into three blocks. Block 1 included the problem-solving MAP scores from the beginning of the semester as well as scores from the metacognitive self-regulation scale of the MSLQ. Block 2 included the Big Five personality trait scores. Block 3 included the scores for the three motivational MSLQ scales.

The residuals of the project score model were skewed, so a Box-Cox transformation was applied to the project score dependent variable (Box and Cox, 1964). The results of this model were similar to results of the original model; however, the transformation made it difficult to interpret the results because the Box-Cox transformation changed the scale of the variable (Osborne, 2002).

3.4.4 RQ3 Data Analysis

In order to examine whether there were reciprocal effects of emotional responses on project outcomes, a special type of Structural Equation Model (SEM) was used: autoregressive cross-lagged models. This type of model was used to examine whether there were reciprocal relationships between emotional responses, self-efficacy, and project score outcomes over the length of the course. The presence of reciprocal relationships provided evidence that students' emotional reactions to programming projects created a feedback loop process.

The autoregressive cross-lagged model was suitable for this analysis because it allows one to examine reciprocal relationships between constructs over time (Selig and Little, 2012). An autoregressive cross-lagged model includes both autoregressive and cross-lagged relationships between variables. Autoregressive relationships are the effects of one variable on later iterations of that same variable, for example, the effects of self-efficacy on later self-efficacy. Cross-lagged effects are the effects of one construct on another construct across time points, for example, the effects of self-efficacy on later programming project scores. The model assumed that constructs affected one another across time points in order to capture the presumed reciprocal relationships between the constructs. By fitting this type of model to the emotional response question data, it could be determined whether there was evidence of reciprocal effects between emotional reactions and programming project scores. Such evidence would support the notion that emotional responses and projects can operate as a feedback loop.

In order to fit the autoregressive cross-lagged model to the emotional response and project data, the data was condensed. This was necessary because of the recommended sample size guidelines for structural equation models, which suggest that 20 observations per estimated parameter is ideal (Kline, 2015). Instead of considering the 10 time points separately, the variables were combined into three time chunks. The project scores and emotional response scores from projects 2,3, and 4, projects 5,6, and 7, and projects 8,9, and 10 were each averaged to create three project score measures and three measures for each of the emotional response questions across the semester (in the model results tables and diagrams in the next chapter, these combined variables are denoted in the following format: frustrated 2,3,4, frustrated 5,6,7, frustrated 8,9,10). The emotional response questions and project scores from project 11 were not included in the model, because of significantly greater non-response on the emotional response questions for that project. The data was condensed in this way to reduce the number of parameters to be estimated in the model such that the available sample size was sufficient to reasonably fit the model. The same analysis was conducted with each of the four emotional reaction questions separately.

3.4.5 RQ4 Data Analysis

The goal of RQ4 was to test the effects of a pilot intervention designed to boost students' self-efficacy. Multiple regression was used to assess the impact of the self-evaluation inter-

vention on project scores and self-efficacy. The impact was analyzed using four outcomes: project scores during the self-evaluation weeks (projects 4-7), self-efficacy during the selfevaluation weeks (projects 4-7), project scores after the intervention weeks (projects 8-11), and self-efficacy after the intervention weeks (projects 8-11). The regressions also controlled for project performance prior to the self-evaluation intervention (projects 1-3), as well as self-efficacy and emotional responses prior to the intervention. This made it possible to determine whether there were short-term effects or longer term effects of completing the self-evaluation task on self-efficacy learning outcomes. T-tests were also used to compare the pre-intervention differences in emotional responses, self-efficacy, and project scores between the intervention group and the control group.

The self-evaluation intervention was offered to over 300 students (half of the class) chosen randomly from the entire class. Of those 300+, 120 agreed to take part in the intervention, and 90 completed the intervention. The impact of the intervention was assessed by comparing the students who completed the intervention to the rest of the students in the class who did not take part, including those who were offered the opportunity to take part in the intervention but declined, as well as those who were not even offered the intervention. The influence of the intervention on those who completed it was assessed using four multiple regression models, one for each of the four outcomes (self-efficacy during intervention, project scores during intervention, self-efficacy after intervention, and project scores after intervention). These multiple regression models also controlled for students' prior project scores, prior self-efficacy, and prior emotional responses.

CHAPTER 4

RESULTS

The first section of this chapter presents summary statistics and plots for the distributions of all the scales used in the analysis. The next section presents results from the analyses used to answer each of the four research questions. These data analyses fall into two main categories: descriptive models (research questions 1-3) and evaluation of intervention (research question 4). The descriptive models portion of the analyses used statistical models to examine the influence of cognitive (problem-solving ability and metacognitive self-regulation), affective (self-efficacy, intrinsic goal orientation, extrinsic goal orientation, and feelings of pride, frustration, and inadequacy), and dispositional (extraversion, agreeableness, conscientiousness, emotional stability, openness to experience) factors on students' course outcomes. Regression models and structural equation models (path analysis and autoregressive cross-lagged models) were used to connect the variables of interest to learning outcomes (project and exam scores). The evaluation of intervention for research question 4 used multiple regression models to assess the impact of the intervention while controlling for prior variables.

4.1 Descriptive Statistics

Descriptive statistics for the various constructs being used in the analyses are presented in this section. This section includes tables of the means, standard deviations, and other descriptive statistics (median, trimmed mean, skew, kurtosis) of the different sets of variables. Histograms and boxplots are also included to show the distributions of some of the variables.

The project data consisted of students' scores on the 11 programming projects (which were worth a varying number of points) that students completed in the semester, as well as the total aggregate project score. The means, standard deviations, and other descriptive statistics for the project scores are shown in table 4.1. Figure 4.1 includes a histogram showing the distribution of total project scores, and a boxplot for each individual project.

	n	mean	sd	median	trimmed	skew	kurtosis
Total Proj. Score	612	370.74	101.02	407	393.48	-2.03	3.60
Proj 1	612	13.80	3.06	15	14.54	-3.84	14.28
Proj 2	612	17.84	4.09	20	18.85	-2.97	9.15
Proj 3	612	15.49	5.64	18	16.63	-1.51	1.50
Proj 4	612	31.95	11.81	38	34.64	-1.67	1.59
Proj 5	612	35.67	14.34	43	38.89	-1.62	1.18
Proj 6	612	37.16	13.15	43	40.42	-1.89	2.24
Proj 7	612	42.00	14.46	49	45.75	-1.97	2.62
Proj 8	612	40.17	15.78	48	43.83	-1.69	1.41
Proj 9	612	47.39	15.47	54	51.89	-2.36	4.16
Proj 10	612	42.56	17.20	50	46.28	-1.62	1.27
Proj 11	612	46.71	17.07	55	51.43	-2.06	2.68

Table 4.1: Project Descriptive Statistics

The exam data consisted of students' scores on the three multiple-choice exams that were given across the semester. The exams were worth a varying number of points. Two sets of descriptive statistics for the exam scores are reported, the raw (sum) scores, and the EAP scores. Note that the EAP scores are standardized. Means, standard deviations and other descriptive statistics for the exam scores are shown in table 4.2. Histograms of EAP scores for each exam are shown in figure 4.2.

Table 4.2: Exam Descriptive Statistics

	n	mean	sd	median	trimmed	skew	kurtosis
Exam 1 Total	605	72.69	13.70	76.00	73.44	-0.57	0.27
Exam 1 EAP	605	0.08	0.88	0.03	0.09	-0.15	-0.53
Exam 2 Total	601	113.74	22.32	115.00	114.89	-0.50	-0.08
Exam 2 EAP	601	0.05	0.91	0.02	0.06	-0.13	-0.43
Exam 3 Total	585	126.92	30.29	130.00	127.82	-0.22	-0.76
Exam 3 EAP	585	0.07	0.94	0.03	0.09	-0.13	-0.69



Figure 4.1: Distribution of Individual and Total Project Scores

The MSLQ data consisted of students' scores on the four MSLQ scales: self-efficacy, intrinsic goal orientation, extrinsic goal orientation, and metacognitive self-regulation. All MSLQ items were seven point Likert items, and total scores consist of sum scores for each scale. Means, standard deviations, and other descriptive statistics for the MSLQ scores are shown in table 4.3.

 Table 4.3: MSLQ Descriptive Statistics

	n	mean	sd	median	trimmed	skew	kurtosis
Metacognitive Self-Regulation	547	46.86	9.18	47	46.90	-0.07	0.18
Self-efficacy	547	42.17	8.93	44	42.59	-0.54	0.06
Intrinsic Goal Orientation	547	19.57	4.66	20	19.74	-0.38	0.03
Extrinsic Goal Orientation	547	22.06	4.47	23	22.37	-0.70	0.48

The Big Five data consisted of students' scores on the five personality trait scales: extraversion, agreeableness, conscientiousness, openness to experience, and stability. All Big



Figure 4.2: Exam Score Distributions

Five items were seven point Likert items, each scale consisted of 10 items, and total scores for each trait consisted of the sum of the 10 items. Means, standard deviations, and other descriptive statistics for the Big Five scores are shown in table 4.4.

	n	mean	sd	median	trimmed	skew	kurtosis
Extraversion	540	40.03	10.32	40	40.10	-0.08	0.05
Agreeableness	524	49.02	8.56	50	49.18	-0.23	-0.34
Conscientiousness	526	46.80	8.33	47	46.74	0.01	-0.17
Emotional Stability	504	43.51	10.13	44	43.90	-0.37	0.08
Openness to Experience	529	47.85	7.99	48	47.65	0.14	-0.43

Table 4.4: Big Five Descriptive Statistics

The problem-solving data consisted of students' scores on the four problem-solving items that were administered at the beginning and end of the course. Each item consisted of multiple questions concerning the same problem situation. Students received a score for each item, and a total score that corresponds to the sum of those individual scores. Descriptive statistics for these raw scores are reported below. Additionally, descriptive statistics are reported for the MAP scores for total problem-solving ability. Note that the MAP scores are standardized. Pre and post scores are reported for each measure. The individual item scores correspond to the items as follows:

- Q1 = Library System
- Q2 = Traveling
- Q3 = Troubleshooting Water Gates
- Q4 = Drawing System

Means, standard deviations, and other descriptive statistics for the problem-solving scores are shown in table 4.5. Histograms of the problem-solving MAP score distributions are shown in figure 4.3.

The emotional response data consisted of student responses to the four emotional response questions between projects 2 and 11. The emotional response questions were ad-

	n	mean	sd	median	trimmed	skew	kurtosis
PS Total Pretest	557	9.50	5.08	9.50	9.41	0.13	-0.69
PS Q1 Pretest	557	1.90	1.67	1.50	1.67	1.06	0.34
PS Q2 Pretest	557	1.08	1.30	1.00	0.86	1.15	0.14
PS Q3 Pretest	557	3.15	2.08	3.33	3.19	-0.02	-1.28
PS Q4 Pretest	557	3.36	2.33	4.00	3.45	-0.16	-1.66
PS MAP Pretest	557	-0.52	1.91	-0.79	-0.63	0.58	0.17
PS Total Posttest	541	10.77	5.92	11.33	10.87	-0.15	-0.89
PS Q1 Posttest	541	2.34	2.08	1.50	2.17	0.56	-1.09
PS Q2 Posttest	541	1.40	1.50	1.00	1.24	0.71	-0.99
PS Q3 Posttest	541	3.49	2.17	4.00	3.61	-0.26	-1.33
PS Q4 Posttest	541	3.55	2.38	5.00	3.68	-0.28	-1.63
PS MAP Posttest	541	0.04	2.26	-0.09	-0.03	0.25	-0.52

Table 4.5: Problem Solving Descriptive Statistics

ministered as seven-point Likert items. The descriptive statistics are divided by question, means, standard deviations and other descriptive statistics are provided below for each of the four questions in tables 4.6, 4.7, 4.8, and 4.9 (across the 10 repeated measures).

	n	mean	sd	median	trimmed	skew	kurtosis
P2 Frustrated	347	4.02	1.78	4	4.00	-0.03	-1.03
P3 Frustrated	324	4.53	1.78	5	4.60	-0.29	-1.02
P4 Frustrated	300	4.61	1.78	5	4.70	-0.27	-0.96
P5 Frustrated	312	4.55	1.92	5	4.66	-0.27	-1.13
P6 Frustrated	328	4.44	1.82	5	4.50	-0.16	-1.07
P7 Frustrated	310	4.19	1.83	4	4.19	0.11	-1.12
P8 Frustrated	269	4.06	1.76	4	4.03	0.11	-1.00
P9 Frustrated	300	3.47	1.70	3	3.37	0.41	-0.60
P10 Frustrated	238	4.47	1.93	5	4.54	-0.13	-1.26
P11 Frustrated	279	3.68	1.76	3	3.60	0.40	-0.76

 Table 4.6:
 Frustration Descriptive Statistics



Figure 4.3: Problem Solving MAP scores: Pretest and Posttest

	n	mean	sd	median	trimmed	skew	kurtosis
P2 Inadequate	347	3.14	1.80	3	2.99	0.43	-0.86
P3 Inadequate	322	3.42	1.89	3	3.28	0.43	-0.87
P4 Inadequate	298	3.74	1.91	4	3.68	0.19	-1.12
P5 Inadequate	313	3.76	1.92	4	3.71	0.21	-1.08
P6 Inadequate	327	3.63	1.90	3	3.54	0.25	-1.09
P7 Inadequate	309	3.46	1.88	3	3.35	0.35	-1.03
P8 Inadequate	269	3.37	1.85	3	3.24	0.39	-0.89
P9 Inadequate	301	2.98	1.71	3	2.80	0.57	-0.53
P10 Inadequate	238	3.76	1.91	4	3.71	0.21	-1.06
P11 Inadequate	279	3.20	1.84	3	3.04	0.52	-0.75

 Table 4.7: Inadequate Descriptive Statistics

4.2 RQ1 Results

The results of the path analysis are presented in this section. Two versions of the path analysis were conducted, for the two outcome variables: total project scores and total

	n	mean	sd	median	trimmed	skew	kurtosis
P2 Self-Efficacy	342	5.36	1.25	6	5.47	-0.86	0.66
P3 Self-Efficacy	321	5.21	1.33	5	5.32	-0.72	0.29
P4 Self-Efficacy	299	5.22	1.36	5	5.36	-0.94	0.91
P5 Self-Efficacy	311	5.02	1.56	5	5.18	-0.79	-0.03
P6 Self-Efficacy	329	5.07	1.49	5	5.21	-0.84	0.21
P7 Self-Efficacy	309	5.20	1.48	6	5.37	-1.02	0.54
P8 Self-Efficacy	269	5.27	1.43	6	5.44	-1.06	0.66
P9 Self-Efficacy	301	5.35	1.34	6	5.49	-0.81	0.01
P10 Self-Efficacy	238	5.23	1.45	6	5.40	-1.01	0.65
P11 Self-Efficacy	278	5.42	1.36	6	5.61	-1.23	1.44

 Table 4.8: Self-efficacy Descriptive Statistics

Table 4.9: Proud Descriptive Statistics

	n	mean	sd	median	trimmed	skew	kurtosis
P2 Proud	342	5.87	1.34	6	6.12	-1.71	3.19
P3 Proud	319	5.68	1.45	6	5.92	-1.31	1.41
P4 Proud	300	5.62	1.52	6	5.88	-1.26	1.17
P5 Proud	309	5.54	1.63	6	5.82	-1.25	0.92
P6 Proud	328	5.59	1.58	6	5.85	-1.22	0.90
P7 Proud	311	5.65	1.44	6	5.89	-1.35	1.77
P8 Proud	268	5.65	1.48	6	5.90	-1.27	1.32
P9 Proud	301	5.64	1.26	6	5.80	-0.90	0.62
P10 Proud	237	5.44	1.64	6	5.70	-1.16	0.84
P11 Proud	278	5.52	1.49	6	5.76	-1.28	1.56

exam scores. For each analysis, first an initial model was fit to the data, then the fit was assessed and modifications to the model were made using modification indices as necessary to achieve a well-fitting model. First, the project score path analysis is presented, followed by the exam score path analysis. Models are estimated using maximum likelihood with robust standard errors and a Satorra-Bentler scaled test statistic to deal with non-normality in the data (Hox and Bechger, 2007).

4.2.1 Project Score Path Analysis

The initial model (see Figure 3.3) was fit first to the overall project scores as the primary outcome. The fit of the initial model is shown in table 4.10 (note that good model fit is indicated by a χ^2 p-value >0.05, and an RMSEA <0.05).

Table 4.10: Project Score Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI Lo	RMSEA CI Up	RMSEA p-value
Initial Model	234.31	26.00	0.00	0.14	0.13	0.16	0.00

The chi-square test and the RMSEA confidence interval indicated that the initial model did not fit the data well (χ^2 (1) = 234.31, p < 0.05, RMSEA = 0.14). The parameter values for the initial model are shown in table 4.11.

DV	IV	Coef.	SE	Ζ	p-value
Total Project Score	Conscientiousness	0.10	0.05	2.12	0.03
Total Project Score	Metacognitive Self-Reg.	-0.05	0.05	-0.94	0.35
Total Project Score	Self-efficacy	0.02	0.05	0.41	0.68
Total Project Score	Problem Solving Pre	0.19	0.05	3.97	0.00
Total Project Score	Openness to Experience	0.04	0.06	0.60	0.55
Problem Solving Pre	Self-efficacy	0.07	0.06	1.26	0.21
Self-efficacy	Metacognitive Self-Reg.	0.15	0.04	4.01	0.00
Self-efficacy	Conscientiousness	0.07	0.04	1.60	0.11
Self-efficacy	Intrin. Goal Orient.	0.47	0.04	11.99	0.00
Self-efficacy	Openness to Experience	0.17	0.05	3.39	0.00
Self-efficacy	Extraversion	-0.08	0.04	-1.82	0.07
Self-efficacy	Extrin. Goal Orient.	0.25	0.04	6.65	0.00
Metacognitive Self-Reg.	Conscientiousness	0.10	0.05	1.85	0.06
Metacognitive Self-Reg.	Openness to Experience	0.22	0.05	4.41	0.00
Intrin. Goal Orient.	Conscientiousness	0.08	0.05	1.51	0.13
Intrin. Goal Orient.	Openness to Experience	0.21	0.06	3.68	0.00
Intrin. Goal Orient.	Agreeableness	0.02	0.06	0.40	0.69
Extrin. Goal Orient.	Agreeableness	0.04	0.05	0.81	0.42

Table 4.11: Standardized Path Coefficients: Project Score Initial Model

Modification indices from the initial model were used to modify the initial model



Figure 4.4: Modified Path Model: Project Scores

so that it would be a better fit for the data. Paths were removed from the initial model, and new paths and covariances were added to the model, based on which had the largest modification indices, and which were conceptually reasonable based on prior work (e.g. personality traits should be hypothesized to influence self-efficacy, and not vice versa). The modified model for project scores added covariances between the MSLQ scales: (covariance between metacognitive self-regulation and intrinsic goal orientation, covariance between metacognitive self-regulation and extrinsic goal orientation, covariance between intrinsic and extrinsic goal orientation), a path from openness to experience to problem solving, a path from extrinsic goal orientation. The paths from self-efficacy to problem solving, from self-efficacy to project scores, and from agreeableness to intrinsic goal orientation were removed because the coefficients were minuscule and insignificant. The path diagram of the modified model is shown in 4.4, with standardized path coefficients shown for statistically significant paths (p<0.05).

 Table 4.12: Project Score Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI Lo	RMSEA CI Up	RMSEA p-value
Initial Model	234.31	26.00	0.00	0.14	0.13	0.16	0.00
Modified Model	25.77	23.00	0.31	0.02	0.00	0.05	0.97

The fit of the modified model is shown in table 4.12 (alongside that of the initial model for comparison). The chi-square test and the RMSEA confidence interval indicate that the modified model was a good fit to the data (χ^2 (1) = 25.77, p > 0.05, RMSEA = 0.02). The results of the likelihood ratio test comparing the fit of the two models is shown in table 4.13 (note: the likelihood ratio test uses the uncorrected χ^2 for each model). The test results indicate that the modified model was a significantly better fit to the data than the initial model. The parameters of the modified model are shown in table 4.14.

Table 4.13: Project Score Model Likelihood Ratio Test

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df.diff	p-value
Modified Model Initial Model	$\begin{array}{c} 23.00\\ 26.00 \end{array}$	$31673.10 \\ 31890.81$	31810.15 32015.77	27.03 250.74	NA 181.33	NA 3.00	NA 0.00

DV	IV	Coef.	SE	Ζ	p-value
Total Project Score	Conscientiousness	0.11	0.05	2.33	0.02
Total Project Score	Metacognitive Self-Reg.	-0.03	0.05	-0.60	0.55
Total Project Score	Problem Solving Pre	0.20	0.05	4.24	0.00
Problem Solving Pre	Openness to Experience	0.23	0.05	5.08	0.00
Self-efficacy	Metacognitive Self-Reg.	0.14	0.04	3.14	0.00
Self-efficacy	Conscientiousness	0.06	0.04	1.74	0.08
Self-efficacy	Intrin. Goal Orient.	0.43	0.04	10.46	0.00
Self-efficacy	Openness to Experience	0.15	0.04	3.94	0.00
Self-efficacy	Extraversion	-0.07	0.04	-1.94	0.05
Self-efficacy	Extrin. Goal Orient.	0.23	0.04	6.24	0.00
Metacognitive Self-Reg.	Conscientiousness	0.12	0.05	2.53	0.01
Metacognitive Self-Reg.	Openness to Experience	0.19	0.05	4.01	0.00
Metacognitive Self-Reg.	Extrin. Goal Orient.	1.08	0.40	2.67	0.01
Intrin. Goal Orient.	Conscientiousness	0.08	0.05	1.62	0.10
Intrin. Goal Orient.	Openness to Experience	0.24	0.05	5.33	0.00
Extrin. Goal Orient.	Agreeableness	0.02	0.03	0.67	0.50
Extrin. Goal Orient.	Emotional Stability	-0.12	0.04	-2.87	0.00
Extrin. Goal Orient.	Extraversion	0.11	0.04	2.70	0.01

Table 4.14: Standardized Path Coefficients: Project Score Modified Model

Overall the results of the path analysis provided a good detailed picture of the way that cognitive, affective, and dispositional factors interacted to influence project outcomes. In the final model, the only factors that had statistically significant paths to aggregate project scores were conscientiousness and problem-solving scores. A notable absence is the path from self-efficacy to project scores, which was removed from the initial model because it was not significant. This is surprising because virtually all of the existing research has suggested that there would be a connection between self-efficacy and learning outcomes.

4.2.2 Exam Score Path Analysis

 Table 4.15: Exam Score Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI Lo	RMSEA CI Up	RMSEA p-value
Initial Model	239.61	26.00	0.00	0.15	0.13	0.16	0.00

Next the initial model was fit to the data with aggregate exam scores as the outcome measure. The model fit of the initial model is shown in table 4.15. As was the case for the project scores outcome, the initial model was not a good fit to the data (χ^2 (1) = 239.61, p < 0.05, RMSEA = 0.15), as indicated by the test statistic and the RMSEA confidence interval. The parameters of the initial model are shown in table 4.16.

Modification indices from the initial model were used to modify the initial model so that it would be a better fit for the data. Paths were removed from the model, and new paths and covariances were added to the model, based on which had the largest modification indices, and which were conceptually reasonable (e.g. personality traits should be hypothesized to influence self-efficacy, and not vice versa). The modified model for project scores added covariances between the MSLQ scales: (covariance between metacognitive selfregulation and intrinsic goal orientation, covariance between metacognitive self-regulation and extrinsic goal orientation, covariance between intrinsic and extrinsic goal orientation), a path from openness to experience to problem solving, a path from extrinsic goal orientation

DV	IV	Coef.	SE	Ζ	p-value
Total Exam Score	Conscientiousness	0.04	0.05	0.82	0.41
Total Exam Score	Metacognitive Self-Reg.	-0.03	0.05	-0.67	0.50
Total Exam Score	Self-efficacy	0.10	0.05	1.98	0.05
Total Exam Score	Problem Solving Pre	0.20	0.05	4.31	0.00
Total Exam Score	Openness to Experience	-0.02	0.05	-0.41	0.68
Problem Solving Pre	Self-efficacy	0.08	0.05	1.65	0.10
Self-efficacy	Metacognitive Self-Reg.	0.14	0.04	3.55	0.00
Self-efficacy	Conscientiousness	0.07	0.04	1.68	0.09
Self-efficacy	Intrin. Goal Orient.	0.46	0.04	12.82	0.00
Self-efficacy	Openness to Experience	0.17	0.04	3.96	0.00
Self-efficacy	Extraversion	-0.08	0.04	-1.95	0.05
Self-efficacy	Extrin. Goal Orient.	0.25	0.04	6.60	0.00
Metacognitive Self-Reg.	Conscientiousness	0.10	0.05	1.90	0.06
Metacognitive Self-Reg.	Openness to Experience	0.21	0.05	4.26	0.00
Intrin. Goal Orient.	Conscientiousness	0.07	0.05	1.42	0.16
Intrin. Goal Orient.	Openness to Experience	0.21	0.05	3.85	0.00
Intrin. Goal Orient.	Agreeableness	0.02	0.05	0.43	0.66
Extrin. Goal Orient.	Agreeableness	0.05	0.05	0.95	0.34

Table 4.16: Standardized Path Coefficients: Exam Score Initial Model

to metacognitive self-regulation, and a path from extraversion to extrinsic goal orientation. The paths from self-efficacy to problem solving, from self-efficacy to project scores, and from agreeableness to intrinsic goal orientation were removed because the coefficients were minuscule and insignificant. The path diagram of the modified model is shown in figure 4.5, with standardized path coefficients indicated for statistically significant paths.

Table 4.17: Exam Score Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI Lo	RMSEA CI Up	RMSEA p-value
Initial Model Modified Model	228.18 28.01	$26.00 \\ 23.00$	0.00 0.22	$0.15 \\ 0.03$	0.13 0.00	0.16 0.05	0.00 0.93

The fit indices for the modified model, alongside those of the initial model for comparison, are shown in table 4.17. The modified model was a good fit to the data based on the test statistic and the RMSEA (χ^2 (1) = 28.01, p > 0.05, RMSEA = 0.03). The results



Figure 4.5: Modified Path Model: Exam Scores

of the likelihood ratio test comparing the fit of the modified model with the initial model are shown in table 4.18 (note: the likelihood ratio test uses the uncorrected χ^2 for each model). The likelihood ratio test results indicate that the modified model was a significantly better fit to the data than the initial model. The parameters of the modified model are shown in table 4.19.

Table 4.18: Exam Score Model Likelihood Ratio Test

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df.diff	p-value
Modified Model	23.00	28081.26	28193.50	29.96	NA	NA	NA
Initial Model	26.00	28294.84	28395.06	249.55	219.59	3.00	0.00

The path analysis results for the exam scores provided a good detailed picture of the way that cognitive, affective, and dispositional factors interacted to influence exam outcomes. In the final model, the only factors that had statistically significant paths to

DV	IV	Coef.	SE	Ζ	p-value
Total Exam Score	Conscientiousness	0.06	0.05	1.16	0.25
Total Exam Score	Problem Solving Pre	0.20	0.05	4.23	0.00
Total Exam Score	Extraversion	-0.12	0.05	-2.53	0.01
Problem Solving Pre	Openness to Experience	0.27	0.05	5.87	0.00
Problem Solving Pre	Metacognitive Self-Reg.	-0.10	0.05	-2.10	0.04
Self-efficacy	Metacognitive Self-Reg.	0.14	0.04	3.15	0.00
Self-efficacy	Intrin. Goal Orient.	0.43	0.04	10.27	0.00
Self-efficacy	Openness to Experience	0.18	0.04	4.63	0.00
Self-efficacy	Extraversion	-0.07	0.04	-1.91	0.06
Self-efficacy	Extrin. Goal Orient.	0.23	0.04	6.12	0.00
Metacognitive Self-Reg.	Conscientiousness	0.12	0.05	2.48	0.01
Metacognitive Self-Reg.	Openness to Experience	0.19	0.05	3.86	0.00
Metacognitive Self-Reg.	Extrin. Goal Orient.	1.01	0.38	2.66	0.01
Intrin. Goal Orient.	Conscientiousness	0.07	0.05	1.48	0.14
Intrin. Goal Orient.	Openness to Experience	0.23	0.05	5.05	0.00
Extrin. Goal Orient.	Agreeableness	0.03	0.04	0.75	0.45
Extrin. Goal Orient.	Emotional Stability	-0.13	0.04	-2.95	0.00
Extrin. Goal Orient.	Extraversion	0.12	0.04	2.71	0.01

Table 4.19: Standardized Path Coefficients: Exam Score Modified Model

aggregate exam scores were extraversion and problem-solving scores. Problem solving scores had the largest significant positive effect on exam scores (p<0.05). Interestingly, extraversion had a statistically significant negative impact on exam scores (p<0.05). A notable absence is the path from self-efficacy to exam scores, which was removed from the initial model because it was not significant. This is surprising because prior research has suggested that there would be a connection between self-efficacy and learning outcomes.

The results of the path analyses answered the question of how cognitive, affective, and dispositional factors interact and influence learning outcomes in programming. The significant paths of the modified models exhibited which relationships hold between these constructs for programming students. Furthermore, the path coefficients in the modified models revealed which factors had the most influence on the two learning outcome variables. Overall, the takeaway is that problem solving ability had the strongest overall effect on student learning outcomes as it was the largest predictor (by Z score) in both modified models, and it was the only significant predictor of both the project and exam score outcomes. Otherwise, the only significant predictors of project or exam scores revealed by these results were conscientiousness, with a significant positive effect on project scores, and extraversion, with a significant negative effect on exam scores. A notable absence was the lack of an effect of self-efficacy on either outcome.

4.3 RQ2 Results

Multiple regression models were fit using all of the student factors as independent variables and each of the aggregate project and exam scores as dependent variables. The model fit statistics for the project scores model are shown in table 4.20. Note that a Box-Cox transformation was applied to the project score variable, so coefficients are not interpretable on the project score scale.

 \mathbf{R}^2 ΔR^2 Adj. \mathbb{R}^2 SE ΔF R DF1 DF2 Sig. p-value *** Block 1 0.22 0.050.0423127.78 0.0510.3724130.00Block 2 0.270.0722972.12 0.06 0.060.022.125408Block 3 0.270.070.0523036.340.000.243 4050.87

Table 4.20: Project Score MR Model Summary Statistics

Block 1 included problem-solving scores and metacognitive self-regulation as the predictors and project scores as outcome. The results indicated the model was significant, and that the variables explained about 5% of the variance in students' project scores (F(2,413) = 10.37, p < 0.001, $\Delta R^2 = 0.05$).

Block 2 added the five Big Five personality variables to the model. The results indicated that the model was not a significant improvement over block 1, and that the big five variables only explained an additional 2% of the variance in project scores (F(5,408) = 2.12, p > 0.05, $\Delta R^2 = 0.02$).

Block 3 added the three other MSLQ variables to the model. The results indicated

that the model was not a significant improvement over blocks 1 and 2, and that the MSLQ variables did not explain any additional variance in project scores (F(3,405) = 0.24, p > 0.05, $\Delta R^2 = 0$).

Overall, the variance explained by the model was low ($\mathbb{R}^2 = 0.07$), and most of the unique variance explained came from block 1. The multiple regression coefficients for the project scores model are shown in table 4.21. The coefficients indicated that problem solving was the strongest significant predictor of project scores (p < 0.05), and that conscientiousness also significantly predicted project scores.

Model	Predictor	Coef.	SE	t-stat.	p-value	Sig.
Model 1	(Intercept)	81058.59	5904.59	13.73	0.00	***
Model 1	Problem Solving Pre	2663.50	587.36	4.53	0.00	***
Model 1	Metacognitive Self-Reg.	-26.04	123.95	-0.21	0.83	
Model 2	(Intercept)	64415.26	10230.55	6.30	0.00	***
Model 2 $$	Problem Solving Pre	2291.75	605.93	3.78	0.00	***
Model 2	Metacognitive Self-Reg.	-91.76	129.77	-0.71	0.48	
Model 2	Extraversion	-161.60	118.25	-1.37	0.17	
Model 2	Agreeableness	-47.94	153.02	-0.31	0.75	
Model 2	Conscientiousness	343.58	152.72	2.25	0.02	*
Model 2	Emotional Stability	71.45	118.89	0.60	0.55	
Model 2	Openness to Experience	188.31	169.00	1.11	0.27	
Model 3	(Intercept)	61793.76	11340.36	5.45	0.00	***
Model 3	Problem Solving Pre	2272.30	609.99	3.73	0.00	***
Model 3	Metacognitive Self-Reg.	-154.05	156.76	-0.98	0.33	
Model 3	Extraversion	-154.36	121.12	-1.27	0.20	
Model 3	Agreeableness	-42.28	154.00	-0.27	0.78	
Model 3	Conscientiousness	338.84	153.53	2.21	0.03	*
Model 3	Emotional Stability	62.92	120.85	0.52	0.60	
Model 3	Openness to Experience	168.03	176.10	0.95	0.34	
Model 3	Self-efficacy	108.37	186.13	0.58	0.56	
Model 3	Intrin. Goal Orient.	28.84	341.88	0.08	0.93	
Model 3	Extrin. Goal Orient.	60.82	283.81	0.21	0.83	

Table 4.21: Project Score MR Model Coefficient Estimates

Next, the same model was fit using the aggregate exam scores as the outcome

	R	\mathbf{R}^2	Adj. \mathbb{R}^2	SE	ΔR^2	ΔF	DF1	DF2	p-value	Sig.
Block 1	0.24	0.06	0.05	58.02	0.06	24.96	1	414	0.00	***
Block 2	0.27	0.08	0.06	57.79	0.02	1.65	5	409	0.15	
Block 3	0.30	0.09	0.07	57.67	0.01	1.41	4	405	0.23	

variable. The model fit statistics for the exam scores model are shown in table 4.22.

 Table 4.22: Exam Score MR Model Summary Statistics

Block 1 included problem-solving and metacognitive self-regulation as the predictors and exam scores as outcome. The results indicated the model was significant, and that the variables explained about 6% of the variance in students' project scores (F(1,414) = 24.96, p < 0.001, $\Delta R^2 = 0.06$).

Block 2 added the five Big Five personality variables to the model. The results indicated that the model was not a significant improvement over block 1, and that the big five variables explained an additional 2% of the variance in exam scores (F(5,409) = 1.65, p > 0.05, $\Delta R^2 = 0.02$).

Block 3 added the three other MSLQ variables to the model. The results indicated that the model was not a significant improvement over blocks 1 and 2, and that the MSLQ variables explained an additional 1% of the variance in exam scores (F(4,405) = 1.41, p > 0.05, R² Change = 0.01).

Overall, the variance explained by the model was low ($\mathbb{R}^2 = 0.09$), and most of the unique variance explained came from block 1. The multiple regression coefficients for the project scores model are shown in table 4.23. The coefficients indicated that only problem solving was a significant predictor of exam scores (p < 0.05).

Next, gender differences in the multiple regression models were examined. To examine this, the models were fit separately for male and female students for both the project score and exam score outcomes. The models were fit using the exact same hierarchical multiple regression models as the previous models above. The coefficients for the male and female models are shown in table 4.24. The project score coefficients show that initial

Model	Predictor	Coef.	SE	t-stat.	p-value	Sig.
Model 1	(Intercept)	321.62	2.90	110.83	0.00	***
Model 1	Problem Solving Pre	7.35	1.47	5.00	0.00	***
Model 2	(Intercept)	326.19	23.96	13.61	0.00	***
Model 2	Problem Solving Pre	7.09	1.51	4.68	0.00	***
Model 2	Extraversion	-0.54	0.30	-1.80	0.07	•
Model 2	Agreeableness	-0.24	0.38	-0.61	0.54	
Model 2	Conscientiousness	0.71	0.38	1.87	0.06	
Model 2	Emotional Stability	-0.11	0.30	-0.38	0.71	
Model 2	Openness to Experience	-0.01	0.42	-0.03	0.98	
Model 3	(Intercept)	339.78	28.39	11.97	0.00	***
Model 3	Problem Solving Pre	6.86	1.53	4.49	0.00	***
Model 3	Extraversion	-0.39	0.30	-1.29	0.20	
Model 3	Agreeableness	-0.17	0.39	-0.45	0.66	
Model 3	Conscientiousness	0.70	0.38	1.81	0.07	
Model 3	Emotional Stability	-0.25	0.30	-0.83	0.41	
Model 3	Openness to Experience	-0.22	0.44	-0.51	0.61	
Model 3	Metacognitive Self-Reg.	-0.58	0.39	-1.49	0.14	
Model 3	Self-efficacy	0.63	0.47	1.34	0.18	
Model 3	Intrin. Goal Orient.	0.81	0.86	0.95	0.34	
Model 3	Extrin. Goal Orient.	-0.97	0.71	-1.36	0.18	

Table 4.23: Exam Score MR Model Coefficient Estimates

problem-solving ability was a statistically significant predictor of male students' project scores, but not for female students' project scores.

Next the models were fit for the exam score outcome. The coefficients for the male and female models are shown in table 4.25. The exam score coefficients show that initial problem-solving ability was a statistically significant predictor for both male and female students' exam scores (p<0.05). Furthermore, the results also show that metacognitive selfregulation was a significant predictor for male students' exam scores, but not for female students' exam scores (p<0.05).

Overall, the multiple regression results suggested that these three sets of factors (problem solving, personality traits, MSLQ characteristics) did not play a large role in explaining students' course outcomes, which is surprising given previous results, which found

Model	Predictor	Male Coef	Male Sig.	Female Coef	Female Sig.
Model 1	(Intercept)	390.53	***	397.28	***
Model 1	Problem Solving Pre	8.80	***	6.81	
Model 2	(Intercept)	359.32	***	268.08	***
Model 2	Problem Solving Pre	7.97	***	5.92	
Model 2	Extraversion	-0.49		-0.17	
Model 2	Agreeableness	-0.76		1.51	
Model 2	Conscientiousness	0.60		0.96	
Model 2	Emotional Stability	0.54		-0.06	
Model 2	Openness to Experience	0.73		0.28	
Model 3	(Intercept)	371.85	***	232.18	**
Model 3	Problem Solving Pre	7.99	***	6.30	
Model 3	Extraversion	-0.42		-0.29	
Model 3	Agreeableness	-0.77		1.51	
Model 3	Conscientiousness	0.65		0.89	
Model 3	Emotional Stability	0.45		0.00	
Model 3	Openness to Experience	0.78		0.23	
Model 3	Metacognitive Self-Reg.	-0.86		0.46	
Model 3	Self-efficacy	-0.19		0.27	
Model 3	Intrin. Goal Orient.	1.50		-0.37	
Model 3	Extrin. Goal Orient.	0.11		0.80	

Table 4.24: Project Score MR Model Coefficient Estimates by Gender

these factors explained significant variance in academic outcomes with this population. The variance explained by these factors was low for both project scores and exam scores, suggesting that there may be other factors that were not examined in this model that could account for more variance in student outcomes. Furthermore, the breakdown by gender showed that the predictors of learning outcomes were similar for male and female students. It was interesting, however, that the results from both regression models found that problem-solving ability was the most important predictor. This aligns with the findings from the path analysis from RQ1. These results also suggest that the two forms of assessment (projects and exam scores) were perhaps more similar than previously thought.

Model	Predictor	Male Coef	Male Sig.	Female Coef	Female Sig.
Model 1	(Intercept)	319.84	***	328.10	***
Model 1	Problem Solving Pre	7.13	***	7.89	**
Model 2	(Intercept)	345.94	***	250.37	***
Model 2	Problem Solving Pre	6.85	***	7.90	*
Model 2	Extraversion	-0.63		-0.32	
Model 2	Agreeableness	-0.74		1.12	
Model 2	Conscientiousness	0.40		1.20	
Model 2	Emotional Stability	0.02		0.12	
Model 2	Openness to Experience	0.31		-0.63	
Model 3	(Intercept)	362.83	***	258.59	***
Model 3	Problem Solving Pre	6.70	***	8.28	*
Model 3	Extraversion	-0.45		-0.31	
Model 3	Agreeableness	-0.70		1.04	
Model 3	Conscientiousness	0.41		0.94	
Model 3	Emotional Stability	-0.12		0.05	
Model 3	Openness to Experience	0.13		-0.94	
Model 3	Metacognitive Self-Reg.	-1.06	*	0.50	
Model 3	Self-efficacy	0.53		0.74	
Model 3	Intrin. Goal Orient.	1.70		-0.31	
Model 3	Extrin. Goal Orient.	-0.87		-1.03	

Table 4.25: Exam Score MR Model Coefficient Estimates by Gender

4.4 RQ3 Results

To investigate the relationships between the emotional response questions and the project scores, separate analyses were conducted for each of the four emotional response questions. Each analysis started with the same initial path model, an autoregressive cross-lagged model, which included covariances between the project scores and the emotional reaction questions within each of the three time chunks (Time chunk 1 = Projects 2,3 & 4; time chunk 2 = Projects 5,6 & 7; time chunk 3 = Projects 8,9 & 10), as well as autoregressive and cross-lagged paths across adjacent time points. Autoregressive are effects of a variable on itself at a later time point, so for example, this model contained autoregressive paths from project scores 2,3,4 to project scores 5,6,7. Cross-lagged paths are effects of a variable on a


Figure 4.6: Frustrated Initial Model

different variable at a later time point, so for example, this model included a cross-lagged path from projects 2,3,4 to an emotional response 5,6,7. In order to say there was evidence for a feedback loop process, the cross-lagged paths needed to be significant. This model specification is hereafter referred to as the "initial model."

4.4.1 Frustrated Questions

First the initial model was fit to the frustrated questions. The model diagram is shown in figure 4.6. The model fit statistics for the autoregressive cross-lagged model on the frustration questions are shown in table 4.26. (Note that good model fit is indicated by a χ^2 p-value greater than 0.05, and an RMSEA below 0.05).

 Table 4.26:
 Frustration Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	RMSEA <= 0.05
Frustrated Model	37.37	4.00	0.00	0.16	0.12	0.21	0.00

The model fit statistics indicate that the initial model was not a good fit to the data (χ^2 (1) = 37.37, p < 0.05, RMSEA = 0.16). The estimated path coefficients are shown in the table 4.27, and estimated covariances are shown in the table 4.28.

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7 Proj Scores: 5,6,7 Frustrated Qs: 5,6,7 Frustrated Qs: 5,6,7 Proj Scores: 8,9,10	Proj Scores: 2,3,4 Frustrated Qs: 2,3,4 Proj Scores: 2,3,4 Frustrated Qs: 2,3,4 Proj Scores: 5,6,7	$\begin{array}{c} 0.37 \\ -0.08 \\ -0.05 \\ 0.59 \\ 0.63 \end{array}$	$\begin{array}{c} 0.05 \\ 0.05 \\ 0.05 \\ 0.04 \\ 0.03 \end{array}$	7.45 -1.45 -1.09 15.41 18.44	$\begin{array}{c} 0.00 \\ 0.15 \\ 0.27 \\ 0.00 \\ 0.00 \end{array}$
Proj Scores: 8,9,10 Frustrated Qs: 8,9,10 Frustrated Qs: 8,9,10	Frustrated Qs: 5,6,7 Proj Scores: 5,6,7 Frustrated Qs: 5,6,7	-0.09 0.01 0.67	$0.04 \\ 0.04 \\ 0.03$	-1.99 0.33 20.65	$0.05 \\ 0.74 \\ 0.00$

 Table 4.27: Frustration Model Estimated Path Coefficients

 Table 4.28:
 Frustration Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Frustrated Qs: 2,3,4	-1.43	0.37	-3.89	0.00
Proj Scores: 5,6,7	Frustrated Qs: 5,6,7	-1.01	0.51	-1.96	0.05
Proj Scores: 8,9,10	Frustrated Qs: 8,9,10	0.79	0.47	1.69	0.09

The estimated path coefficients indicate that by and large, the cross-lagged effects of projects on frustration and frustration on projects were not statistically significant, whereas the auto-regressive effects of earlier projects on later projects, and earlier frustration on later frustration, were significant. The lone exception to this was the path from frustration on projects 5,6, & 7 to scores on projects 8,9, & 10. Overall, these results did not provide evidence for a feedback loop type pattern between students' frustration and project performance.

The estimated covariances indicate the extent to which frustration and project scores were contemporaneously related. These estimated values exhibited an interesting pattern, as the covariance between frustration and projects decreased from the first time chunk to the second time chunk, and the sign actually flipped from the second time chunk to the third time chunk, although this value was not statistically significant. This pattern suggests that frustration was associated with performance significantly at the beginning of the course but that this association dissipated over the course of the semester.



Figure 4.7: Frustrated Modified Model

Because the initial model did not fit the data well, a second specification was fit. The modification indices for the initial model indicated that paths from the project scores at the first time chunk to project scores at the third time chunk, and frustration scores at the first time chunk to frustration scores at the third time chunk, should be added to the model. These additional "long-term" autoregressive effects were added to the initial specification and fit to the data. This model is hereafter referred to as the "modified model." The path diagram for the modified model is shown in figure 4.7.

Table 4.29: Frustration Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	$\mathrm{RMSEA} <= 0.05$
Frustration Mod. Model	1.95	2.00	0.38	0.00	0.00	0.11	0.61

The fit of the modified model is shown in table 4.29. The test statistic and the RMSEA confidence interval indicate that the fit of the modified model to the data was quite good (χ^2 (1) = 1.95, p > 0.05, RMSEA = 0.00). The modified model was compared to the initial model using the likelihood ratio test. The likelihood ratio test results are shown in table 4.30.

The likelihood ratio test indicates that the modified model was a significant im-

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df diff	p-value
Modified Model Initial Model	$2.00 \\ 4.00$	9104.79 9136.21	9175.85 9199.78	$1.95 \\ 37.37$	NA 35.42	NA 2.00	NA 0.00

Table 4.30: Frustration Model Likelihood Ratio Test

provement in fit over the initial model. This suggests that frustration and project scores significantly impacted later iterations of these measures, not just from one time chunk to the next, but across the length of the course. The estimated parameters for the modified model are shown in the two tables below. The estimated path coefficients are shown in table 4.31, and the estimated covariances are shown in table 4.32.

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.37	0.05	7.45	0.00
Proj Scores: 5,6,7	Frustrated Qs: 2,3,4	-0.08	0.05	-1.45	0.15
Frustrated Qs: 5,6,7	Proj Scores: 2,3,4	-0.05	0.05	-1.09	0.27
Frustrated Qs: 5,6,7	Frustrated Qs: 2,3,4	0.59	0.04	15.41	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.57	0.04	14.37	0.00
Proj Scores: 8,9,10	Frustrated Qs: 5,6,7	-0.07	0.04	-1.56	0.12
Proj Scores: 8,9,10	Proj Scores: 2,3,4	0.18	0.05	3.91	0.00
Frustrated Qs: 8,9,10	Proj Scores: 5,6,7	0.02	0.04	0.60	0.55

Table 4.31: Frustration Modified Model Parameter Estimates

Table 4.32: Frustration Modified Model Estimated Covariances

0.53

0.23

0.05

0.05

11.38

4.63

0.00

0.00

Frustrated Qs: 5,6,7

Frustrated Qs: 2,3,4

Frustrated Qs: 8,9,10

Frustrated Qs: 8,9,10

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Frustrated Qs: 2,3,4	-1.43	0.37	-3.89	0.00
Proj Scores: 5,6,7	Frustrated Qs: 5,6,7	-1.01	0.51	-1.96	0.05
Proj Scores: 8,9,10	Frustrated Qs: 8,9,10	0.67	0.44	1.50	0.13

The estimated path coefficients for the modified model were similar to those from the initial path model, with the exception that the two new paths (from project 2,3,4 to



Figure 4.8: Proud Initial Model

project 8,9,10, and from frustrated 2,3,4 to frustrated 8,9,10) were both significant, and the path from frustrated 5,6,7 to project 8,9,10 was no longer significant. The estimated covariances show the same pattern as in the initial model, indicating that the effects of frustration dissipated over the course of the semester.

4.4.2 Proud Questions

Next the model was fit to the proud questions. The initial model is shown in figure 4.8. The model fit statistics for the autoregressive cross-lagged model on the proud questions are shown in table 4.33. Note that good model fit is indicated by a χ^2 p-value greater than 0.05, and an RMSEA below 0.05.

 Table 4.33: Proud Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	RMSEA ≤ 0.05
Proud Model	24.69	4.00	0.00	0.13	0.08	0.18	0.00

The model fit statistics indicate that the model was not a good fit to the data (χ^2 (1) = 24.69, p < 0.05, RMSEA = 0.13). Estimated path coefficients are shown in table 4.34,

and estimated covariances are shown in the table 4.35.

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.37	0.05	7.62	0.00
Proj Scores: 5,6,7	Proud Qs: $2,3,4$	0.12	0.05	2.25	0.02
Proud Qs: 5,6,7	Proj Scores: 2,3,4	0.04	0.05	0.71	0.48
Proud Qs: 5,6,7	Proud Qs: $2,3,4$	0.49	0.04	11.42	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.63	0.04	16.95	0.00
Proj Scores: 8,9,10	Proud Qs: 5,6,7	0.05	0.05	1.08	0.28
Proud Qs: 8,9,10	Proj Scores: 5,6,7	-0.05	0.05	-1.11	0.27
Proud Qs: 8,9,10	Proud Qs: $5,6,7$	0.66	0.04	17.65	0.00

Table 4.34: Proud Model Estimated Path Coefficients

 Table 4.35:
 Proud Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Proud Qs: $2,3,4$	0.70	0.27	2.57	0.01
Proj Scores: 5,6,7	Proud Qs: 5,6,7	2.47	0.44	5.60	0.00
Proj Scores: 8,9,10	Proud Qs: 8,9,10	1.19	0.38	3.16	0.00

The estimated path coefficients indicate that by and large, the cross-lagged effects of projects on pride and pride on projects were not statistically significant, whereas the autoregressive effects of earlier projects on later projects, and earlier pride on later frustration were significant. The lone exception to this was the path from pride on projects 2,3, & 4 to scores on projects 5,6, & 7. Overall, these results do not provide evidence for a feedback loop type pattern where students' feelings of pride influenced their project performance and vice versa.

The estimated covariances indicate the extent to which pride and project scores were contemporaneously related. These estimated values show an interesting pattern, as the covariance between pride and projects increased by a factor of three from the first time point to the second time point, and decreased again by half from the second time point to the third



Figure 4.9: Proud Modified Model

time point. This pattern suggests that pride was associated with performance throughout the semester but that this association was strongest in the middle of the semester.

Because the hypothesized model did not fit the data well, a second specification was fit. The modification indices for the initial model indicated that paths from the project scores at the first time point to project scores at the third time point, and pride scores at the first time point to pride scores at the third time point, should be added to the model. These additional autoregressive effects were added to the initial specification and fit to the data. The path diagram for the modified model is shown in figure 4.9. The fit of the modified model is shown in table 4.36.

Table 4.36: Proud Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	$\mathrm{RMSEA} <= 0.05$
Proud Mod. Model	2.49	2.00	0.29	0.03	0.00	0.12	0.53

The test statistic and the RMSEA confidence interval indicate that the modified model fit the data quite well (χ^2 (1) = 2.49, p > 0.05, RMSEA = 0.03). Next the likelihood ratio test was used to compare the fit of the modified model to the initial model. The results are shown table 4.37.

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df diff	p-value
Modified Model Initial Model	$2.00 \\ 4.00$	8645.09 8663.30	8716.15 8726.87	$2.49 \\ 24.69$	NA 22.20	NA 2.00	NA 0.00

Table 4.37: Proud Model Likelihood Ratio Test

The likelihood ratio test indicates that the modified model is a significant improvement in fit over the initial model. This suggests that pride and project scores significantly impacted later iterations of these measures over time, not just from one time chunk to the next, but across the length of the course. The estimated parameters for the modified model are shown in the two tables below. The estimated path coefficients are shown in table 4.38, and the estimated covariances are shown in table 4.39.

Table 4.38: Proud Modified Model Estimated Path Coefficients

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.37	0.05	7.62	0.00
Proj Scores: 5,6,7	Proud Qs: $2,3,4$	0.12	0.05	2.25	0.02
Proud Qs: 5,6,7	Proj Scores: 2,3,4	0.04	0.05	0.71	0.48
Proud Qs: 5,6,7	Proud Qs: $2,3,4$	0.49	0.04	11.42	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.56	0.04	13.09	0.00
Proj Scores: 8,9,10	Proud Qs: $5,6,7$	0.06	0.04	1.26	0.21
Proj Scores: 8,9,10	Proj Scores: 2,3,4	0.18	0.04	4.10	0.00
Proud Qs: 8,9,10	Proj Scores: 5,6,7	-0.05	0.05	-1.09	0.28
Proud Qs: 8,9,10	Proud Qs: $5,6,7$	0.59	0.05	12.88	0.00
Proud Qs: 8,9,10	Proud Qs: $2,3,4$	0.13	0.05	2.57	0.01

Table 4.39: Proud Modified Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Proud Qs: $2,3,4$	0.70	0.27	2.57	0.01
Proj Scores: 5,6,7	Proud Qs: 5,6,7	2.47	0.44	5.60	0.00
Proj Scores: 8,9,10	Proud Qs: 8,9,10	1.03	0.36	2.84	0.00

The estimated path coefficients for the modified model were similar to those from the initial path model, with the exception that the two new paths (from project 2,3,4 to



Figure 4.10: Inadequate Initial Model

project 8,9,10 and from proud 2,3,4 to proud 8,9,10) are both significant. Once again, the model results suggest that the autoregressive effects were much stronger than the cross-lagged effects, and this does not provide evidence of a feedback loop between feelings of pride and project performance. The estimated covariances showed the same pattern as in the initial model, indicating that the connection between pride and performance was strongest in the middle of the semester.

4.4.3 Inadequate Questions

Next the model was fit to the inadequate questions. The initial model used the same specification as the previous models, which is shown in figure 4.10. The model fit statistics for the autoregressive cross-lagged model on the inadequate questions are shown in table 4.40. Note that good model fit is indicated by a χ^2 p-value greater than 0.05, and an RMSEA below 0.05.

Table 4.40: Inadequate Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	RMSEA <= 0.05
Inadequate Mod	31.35	4.00	0.00	0.15	0.10	0.20	0.00

The model fit statistics indicate that the initial model was not a good fit to the data (χ^2 (1) = 31.35, p < 0.05, RMSEA = 0.15). Estimated path coefficients are shown in table 4.41, and estimated covariances are shown in table 4.42.

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.38	0.05	7.57	0.00
Proj Scores: 5,6,7	Inadequate Qs: 2,3,4	-0.04	0.05	-0.76	0.45
Inadequate Qs: 5,6,7	Proj Scores: 2,3,4	-0.05	0.04	-1.22	0.22
Inadequate Qs: 5,6,7	Inadequate Qs: 2,3,4	0.67	0.03	20.89	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.63	0.03	18.33	0.00
Proj Scores: 8,9,10	Inadequate Qs: 5,6,7	-0.08	0.04	-1.72	0.08
Inadequate Qs: 8,9,10	Proj Scores: 5,6,7	-0.02	0.04	-0.40	0.69
Inadequate Qs: 8,9,10	Inadequate Qs: 5,6,7	0.77	0.02	30.99	0.00

 Table 4.41: Inadequate Model Estimated Path Coefficients

 Table 4.42: Inadequate Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Inadequate Qs: 2,3,4	-1.59	0.40	-4.01	0.00
Proj Scores: 5,6,7	Inadequate Qs: 5,6,7	-1.38	0.51	-2.71	0.01
Proj Scores: 8,9,10	Inadequate Qs: 8,9,10	0.33	0.42	0.79	0.43

The estimated path coefficients indicate that the cross-lagged effects of projects on feelings of inadequacy and feelings of inadequacy on projects were not statistically significant, whereas the autoregressive effects of earlier projects on later projects, and earlier feelings of inadequacy on later feelings of inadequacy were significant. Overall, these results did not provide evidence for a feedback loop type pattern between students' inadequacy and project performance.

The estimated covariances indicate the extent to which feelings of inadequacy and project scores were contemporaneously related. These estimated values show an interesting pattern, as the covariance between feelings of inadequacy and projects decreased from the first time chunk to the second time chunk, and the sign actually flipped from the second



Figure 4.11: Inadequate Modified Model

time chunk to the third time chunk, although this value was not statistically significant. This pattern suggests that inadequacy was associated with performance significantly at the beginning of the course but that this association dissipated over the course of the semester.

Because the hypothesized model did not fit the data well, a second specification was fit. The modification indices for the fit model indicated that paths from the project scores at the first time point to project scores at the third time point, and inadequacy scores at the first time point to inadequacy scores at the third time point, should be added to the model. These additional autoregressive effects were added to the initial model and the modified model was fit to the data. The path diagram for the modified model is shown in figure 4.11. The fit of the modified model is shown in table 4.43.

 Table 4.43: Inadequate Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	$\mathrm{RMSEA} <= 0.05$
Inadequate Mod. Model	3.33	2.00	0.19	0.05	0.00	0.13	0.41

The test statistic and the RMSEA confidence interval indicate that the model was a good fit to the data (χ^2 (1) = 3.33, p > 0.05, RMSEA = 0.05). Next the likelihood ratio test was used to compare the fit of the modified model to that of the initial model. The results are shown in table 4.44.

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df diff	p-value
Modified Model	2.00	9040.65	9111.65	3.33	NA	NA	NA
Initial Model	4.00	9064.68	9128.20	31.35	28.02	2.00	0.00

Table 4.44: Inadequate Model Likelihood Ratio Test

The likelihood ratio test indicates that the modified model was a significant improvement in fit over the initial model. This suggests that inadequacy and project scores significantly impacted later iterations of these measures over time, not just from one time chunk to the next, but across the length of the course. The estimated path coefficients are shown in table 4.45, and the estimated covariances are shown in table 4.46.

Table 4.45: Inadequate Modified Model Estimated Path Coefficients

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.38	0.05	7.57	0.00
Proj Scores: 5,6,7	Inadequate Qs: 2,3,4	-0.04	0.05	-0.76	0.45
Inadequate Qs: 5,6,7	Proj Scores: 2,3,4	-0.05	0.04	-1.22	0.22
Inadequate Qs: 5,6,7	Inadequate Qs: 2,3,4	0.67	0.03	20.89	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.57	0.04	14.39	0.00
Proj Scores: 8,9,10	Inadequate Qs: 5,6,7	-0.05	0.04	-1.21	0.23
Proj Scores: 8,9,10	Proj Scores: 2,3,4	0.18	0.05	3.86	0.00
Inadequate Qs: 8,9,10	Proj Scores: 5,6,7	-0.02	0.04	-0.49	0.62
Inadequate Qs: 8,9,10	Inadequate Qs: 5,6,7	0.64	0.04	14.49	0.00
Inadequate Qs: 8,9,10	Inadequate Qs: 2,3,4	0.18	0.05	3.73	0.00

Table 4.46: Inadequate Modified Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4 Proj Scores: 5,6,7	Inadequate Qs: 2,3,4 Inadequate Qs: 5,6,7	-1.59 -1.38	$\begin{array}{c} 0.40\\ 0.51 \end{array}$	-4.01 -2.71	$\begin{array}{c} 0.00\\ 0.01 \end{array}$
Proj Scores: 8,9,10	Inadequate Qs: 8,9,10	0.41	0.40	1.02	0.31

The estimated path coefficients for the modified model were similar to those from the initial path model, with the exception that the two new paths (from project 2,3,4 to



Figure 4.12: Self-efficacy Initial Model

project 8,9,10 and from inadequate 2,3,4 to inadequate 8,9,10) were both significant. Once again, the model results suggest that the autoregressive effects were much stronger than the cross-lagged effects, and this does not provide evidence of a feedback loop between feelings of inadequacy and project performance. The estimated covariances showed the same pattern as in the initial model, indicating that the connection between inadequacy and performance dissipated over the course of the semester.

4.4.4 Self-efficacy Questions

Lastly the model was fit to the self-efficacy (will do well question) data. The initial model used the same specification as the previous models, which is shown in figure 4.12. The model fit statistics for the autoregressive cross-lagged model on the self-efficacy data are shown in table 4.47. Note that good model fit is indicated by a χ^2 p-value greater than 0.05, and an RMSEA below 0.05.

Table 4.47: Self-efficacy Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	$\mathrm{RMSEA} <= 0.05$
Self-efficacy Model	19.90	4.00	0.00	0.11	0.07	0.17	0.01

The model fit statistics indicate that the initial model was not a good fit to the data (χ^2 (1) = 19.90, p < 0.05, RMSEA = 0.11). Estimated path coefficients are shown in table 4.48, and estimated covariances are shown in table 4.49.

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.36	0.05	7.27	0.00
Proj Scores: 5,6,7	Self-Efficacy Qs: 2,3,4	0.13	0.05	2.49	0.01
Self-Efficacy Qs: 5,6,7	Proj Scores: 2,3,4	0.18	0.05	3.87	0.00
Self-Efficacy Qs: 5,6,7	Self-Efficacy Qs: 2,3,4	0.55	0.04	14.23	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.62	0.04	16.38	0.00
Proj Scores: 8,9,10	Self-Efficacy Qs: 5,6,7	0.07	0.05	1.51	0.13
Self-Efficacy Qs: 8,9,10	Proj Scores: 5,6,7	0.17	0.04	4.54	0.00
Self-Efficacy Qs: 8,9,10	Self-Efficacy Qs: 5,6,7	0.71	0.03	23.99	0.00

Table 4.48: Self-efficacy Model Estimated Path Coefficients

 Table 4.49:
 Self-efficacy
 Model
 Estimated
 Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Self-Efficacy Qs: 2,3,4	0.89	0.26	3.41	0.00
Proj Scores: 5,6,7	Self-Efficacy Qs: 5,6,7	1.96	0.43	4.61	0.00
Proj Scores: 8,9,10	Self-Efficacy Qs: 8,9,10	-0.10	0.32	-0.31	0.76

The estimated path coefficients indicate that three of the four cross-lagged effects between self-efficacy and project scores were statistically significant (the one insignificant path being from self-efficacy 5,6,7 to project 8,9,10), and all four of the autoregressive effects of earlier projects on later projects, and earlier self-efficacy on later self-efficacy, were statistically significant. Overall these results provide evidence of some reciprocal effects between self-efficacy and project scores.

The estimated covariances indicate the extent to which self-efficacy and project scores were contemporaneously related. These estimated values showed an interesting pattern, as the covariance between self-efficacy and projects increased from the first time point to the second time point, and decreased to almost nothing between the second time point to



Figure 4.13: Self-efficacy Modified Model

the third time point. This pattern suggests that self-efficacy was associated with performance through the first two-thirds of the semester before dropping off, and that this association was strongest in the middle of the semester.

Because the hypothesized model did not fit the data well, a second specification was fit. The modification indices for the fit model indicated that paths from the project scores at the first time chunk to project scores at the third time chunk, and self-efficacy scores at the first time chunk to self-efficacy scores at the third time chunk. These additional autoregressive effects were added to the initial specification and fit to the data. The path diagram for the modified model is shown in figure 4.13.

Table 4.50: Self-efficacy Modified Model Fit Statistics

	Test Stat.	df	p-value (χ^2)	RMSEA	RMSEA CI lo	RMSEA CI up	$\mathrm{RMSEA} <= 0.05$
Self-efficacy Mod. Model	1.64	2.00	0.44	0.00	0.00	0.11	0.66

The fit of the modified model is shown in table 4.50. The test statistic and the RMSEA confidence interval indicate that the model was a good fit to the data (χ^2 (1) = 1.64, p > 0.05, RMSEA = 0.00). Next the likelihood ratio test was used to compare the fit of the modified model to that of the initial model. The results are shown in table 4.51.

The likelihood ratio test indicates that the modified model was a significant im-

	Df	AIC	BIC	χ^2	$\Delta \chi^2$	Df diff	p-value
Modified Model	$2.00 \\ 4.00$	8422.96	8493.83	1.64	NA	NA	NA
Initial Model		8437.22	8500.63	19.90	18.26	2.00	0.00

Table 4.51: Self-efficacy Model Likelihood Ratio Test

provement in fit over the initial model. This suggests that self-efficacy and project scores significantly impacted themselves over time, not just from one time chunk to the next, but across the length of the course. The estimated parameters for the modified model are shown in the two tables below. The estimated path coefficients are shown in table 4.52, and the estimated covariances are shown in table 4.53.

Table 4.52: Self-efficacy Modified Model Estimated Path Coefficients

DV	IV	Coef.	SE	Ζ	p-value
Proj Scores: 5,6,7	Proj Scores: 2,3,4	0.36	0.05	7.27	0.00
Proj Scores: 5,6,7	Self-Efficacy Qs: 2,3,4	0.13	0.05	2.49	0.01
Self-Efficacy Qs: 5,6,7	Proj Scores: 2,3,4	0.18	0.05	3.87	0.00
Self-Efficacy Qs: 5,6,7	Self-Efficacy Qs: 2,3,4	0.55	0.04	14.23	0.00
Proj Scores: 8,9,10	Proj Scores: 5,6,7	0.57	0.04	13.51	0.00
Proj Scores: 8,9,10	Self-Efficacy Qs: 5,6,7	0.04	0.05	0.91	0.36
Proj Scores: 8,9,10	Proj Scores: 2,3,4	0.17	0.05	3.79	0.00
Self-Efficacy Qs: 8,9,10	Proj Scores: 5,6,7	0.17	0.04	4.62	0.00
Self-Efficacy Qs: 8,9,10	Self-Efficacy Qs: 5,6,7	0.65	0.04	16.24	0.00
Self-Efficacy Qs: 8,9,10	Self-Efficacy Qs: 2,3,4	0.09	0.04	2.10	0.04

Table 4.53: Self-efficacy Modified Model Estimated Covariances

Var 1	Var 2	Cov.	SE	Ζ	p-value
Proj Scores: 2,3,4	Self-Efficacy Qs: 2,3,4	0.89	0.26	3.41	0.00
Proj Scores: 5,6,7	Self-Efficacy Qs: 5,6,7	1.96	0.43	4.61	0.00
Proj Scores: 8,9,10	Self-Efficacy Qs: 8,9,10	-0.13	0.31	-0.41	0.68

The estimated path coefficients for the modified model were similar to those from the initial path model, with the exception that the two new paths (from project 2,3,4 to project 8,9,10 and from self-efficacy 2,3,4 to self-efficacy 8,9,10) were both significant. Once again, three of the four cross-lagged effects were statistically significant (from project 2,3,4 to self-efficacy 5,6,7; from project 5,6,7 to self-efficacy 8,9,10; and from self-efficacy 2,3,4 to project 5,6,7), which provides evidence of some reciprocal effects between self-efficacy and project scores. The estimated covariances showed the same pattern as in the initial model, indicating that the association between self-efficacy and performance peaked at the middle of the semester and dissipated by the end.

Overall the results of the analyses to answer RQ3 gave a fairly clear answer to the question of how students' emotional reactions relate to their course performance, looking specifically at programming projects. The autoregressive cross-lagged model was used to test for the evidence of reciprocal effects over time between students' individual emotional reactions and their programming performance. With the exception of the self-efficacy (will do well) questions, the analyses did not find any significant evidence for such effects. The autoregressive effects were larger than the cross-lagged effects, with the latter being not statistically significant. In the case of the self-efficacy questions, still only three of the four cross-lagged effects were statistically significant, suggesting that there were some reciprocal effects, but that these perhaps taper off towards the end of the semester.

The estimated covariances provide the remainder of the picture, showing the relationship between the emotional reactions and the project scores at the same time. An interesting pattern of results with the covariances emerged, as there seemed to be different patterns for the positive and negative valence emotional reactions. The two negative emotions (feelings of frustration and inadequacy) showed a similar pattern in which the covariance between the emotional reaction and project scores was the highest at the beginning of the course, and diminished to be statistically insignificant by the end of the course. The two positive emotions (feelings of pride and self-efficacy) showed a similar pattern in which the covariance between the emotional reaction and project scores peaked in the middle of the course. These results suggest that emotional reactions were most strongly associated to the outcomes of projects associated with the same time chunk, and effects on later projects were not significant, with the exception of self-efficacy. This suggests that perhaps, while the projects can cause strong emotional reactions over time, these reactions dissipate without creating any sustained effects.

4.5 RQ4 Results

Research question four examined the impact of the self-evaluation intervention on students' self-efficacy and project scores. First the unconditional differences between the self-evaluation group and the remaining students were examined to assess prior differences between the two groups on project scores, self-efficacy, pride, frustration, and inadequacy. Each variable was further separated into three variables labelled before, during, and after. Before refers to the aggregate of observed values for students' project scores or emotional responses from projects 1-3, which were the projects that were completed before the intervention began. During refers to the aggregate of observed values for students' project scores or emotional responses from projects 4-7, which were the projects that were completed in conjunction with the self-evaluation intervention. After refers to the aggregate of observed values for students' project scores or emotional responses from projects 8-11, which were the projects that were completed after the intervention concluded. T-test results are presented in table 4.54 indicating whether the unconditional differences between the intervention group and the control group were statistically significant before during and after the intervention.

The comparisons in table 4.54 show that the intervention group had significantly higher project scores before (Mean Diff = 6.16, t = 5.60, p < 0.001), during (Mean Diff = 29.83, t = 6.32, p < 0.001), and after (Mean Diff = 30.71, t = 4.87, p < 0.001) the intervention. The intervention group also had significantly higher self-efficacy scores before (Mean Diff = 2.57, t = 4.82, p < 0.001), during (Mean Diff = 4.16, t = 4.27, p < 0.001), and after (Mean Diff = 4.08, t = 3.78, p < 0.001) the intervention. The intervention group also had significantly higher pride scores, before (Mean Diff = 2.37, t = 4.16, p < 0.001), during the intervention.

	Mean Diff.	SE	t-stat.	p-value	Sig
Project Scores Before	6.16	1.10	5.60	0.00	***
Project Scores During	29.83	4.72	6.32	0.00	***
Project Scores After	30.71	6.31	4.87	0.00	***
Self-efficacy Before	2.57	0.53	4.82	0.00	***
Self-efficacy during	4.16	0.97	4.27	0.00	***
Self-efficacy after	4.08	1.08	3.78	0.00	***
Proud Before	2.37	0.57	4.16	0.00	***
Proud During	3.79	1.05	3.63	0.00	***
Proud After	3.70	1.12	3.31	0.00	***
Frustrated Before	0.69	0.46	1.49	0.14	
Frustrated During	1.36	0.89	1.53	0.13	
Frustrated After	1.15	0.84	1.38	0.17	
Inadequate Before	0.68	0.40	1.69	0.09	
Inadequate During	0.74	0.78	0.94	0.35	
Inadequate After	0.46	0.77	0.60	0.55	

Table 4.54: Intervention Completers vs Non-completers: Pre-intervention Mean Differences

(Mean Diff = 3.79, t = 3.63, p < 0.001), and after (Mean Diff = 3.70, t = 3.31, p < 0.001) the intervention. The intervention group's frustrated scores were not significantly different than those of the control group before (Mean Diff = 0.69, t = 1.49, p > 0.05), during (Mean Diff = 1.36, t = 1.53, p > 0.05), or after (Mean Diff = 1.15, t = 1.38, p > 0.05) the intervention. The intervention group's inadequate scores were not significantly different than those of the control group before (Mean Diff = 0.68, t = 1.69, p > 0.05), during (Mean Diff = 0.74, t = 0.94, p > 0.05), or after (Mean Diff = 0.46, t = 0.60, p > 0.05) the intervention.

The multiple regression models below attempted to quantify the impact of the self-evaluation intervention by controlling for prior values of project scores, self-efficacy, and emotional responses. Two outcome variables were considered in these models: self-efficacy and project scores. Effects of the intervention on these outcomes were also considered at two different points, while the intervention was taking place, and after it had concluded. This makes for a total of four models. The models for the two outcomes during the intervention controlled for the before values of project scores, self-efficacy, and emotional response scores.

The models for the two outcomes after the intervention controlled for the before and during values of project scores, self-efficacy, and emotional response scores.

Table 4.55 shows the effects of the self-evaluation intervention on project scores during the intervention, including all the controls. These results exhibited that after controlling for prior values of self-efficacy, project scores, and the other three emotional response questions, those who completed the self-evaluation intervention had significantly higher project scores during the intervention than the control group ($\beta = 12.56$, t = 3.20, p < 0.001). The adjusted R² value of this model was 0.405.

Coef.	SE	t-stat.	p-value	Sig
24.290	6.747	3.600	0.000	***
0.435	0.745	0.584	0.559	
2.525	0.150	16.794	0.000	***
0.455	0.806	0.565	0.572	
0.149	0.867	0.172	0.863	
-0.826 12.555	$0.881 \\ 3.924$	-0.937 3.199	$0.349 \\ 0.001$	***
	Coef. 24.290 0.435 2.525 0.455 0.149 -0.826 12.555	Coef.SE24.2906.7470.4350.7452.5250.1500.4550.8060.1490.867-0.8260.88112.5553.924	Coef.SEt-stat.24.2906.7473.6000.4350.7450.5842.5250.15016.7940.4550.8060.5650.1490.8670.172-0.8260.881-0.93712.5553.9243.199	Coef.SEt-stat.p-value24.2906.7473.6000.0000.4350.7450.5840.5592.5250.15016.7940.0000.4550.8060.5650.5720.1490.8670.1720.863-0.8260.881-0.9370.34912.5553.9243.1990.001

Table 4.55: Intervention Effects: Project Scores During the Intervention

Table 4.56 shows the effects of the self-evaluation intervention on self-efficacy scores during the intervention, including all the controls. These results exhibited that, after controlling for prior values of self-efficacy, project scores, and the other three emotional response questions, those who completed the self-evaluation intervention had self-efficacy scores during the intervention that were not significantly different than those of the control group (β = 0.77, t = 0.97, p > 0.05). The adjusted R² value of this model was 0.403.

Table 4.57 shows the effects of the self-evaluation intervention on project scores after the intervention, including all the controls. These results exhibited that, controlling for previously observed values of self-efficacy, project scores, and the other three emotional response questions, those who completed the self-evaluation intervention had project scores after the intervention that were not significantly different than the control group ($\beta = -1.64$,

	Coef.	SE	t-stat.	p-value	Sig
Intercept	-0.725	1.368	-0.530	0.596	
Self-Efficacy Before	0.939	0.151	6.213	0.000	***
Project Scores Before	0.110	0.030	3.607	0.000	***
Pride Before	0.181	0.163	1.110	0.267	
Inadequate Before	-0.241	0.176	-1.373	0.170	
Frustrated Before Completed Self-Evaluation	$0.051 \\ 0.771$	$0.179 \\ 0.796$	$0.286 \\ 0.969$	$0.775 \\ 0.333$	

Table 4.56: Intervention Effects: Self-efficacy Scores During the Intervention

t = -0.40, p > 0.05). The adjusted R^2 value of this model was 0.642.

 Table 4.57: Intervention Effects: Project Scores after the Intervention

	Coef.	SE	t-stat.	p-value	Sig
Intercept	13.796	7.039	1.960	0.050	*
Self-Efficacy Before	0.077	0.867	0.089	0.929	
Project Scores Before	0.284	0.189	1.506	0.133	
Pride Before	0.109	0.882	0.124	0.902	
Inadequate Before	-0.697	0.990	-0.704	0.481	
Frustrated Before	0.531	0.988	0.537	0.592	
Self-efficacy During	-0.249	0.507	-0.491	0.624	
Project Scores During	1.015	0.044	22.896	0.000	***
Frustrated During	0.079	0.579	0.137	0.891	
Inadequate During	-0.156	0.555	-0.282	0.778	
Pride During	0.267	0.528	0.505	0.614	
Completed Self-Evaluation	-1.637	4.056	-0.404	0.687	

Table 4.58 shows the effects of the self-evaluation intervention on self-efficacy scores after the intervention, including all the controls. These results exhibited that, controlling for previously observed values of self-efficacy, project scores, and the other three emotional response questions, those who completed the self-evaluation intervention had self-efficacy scores after the intervention that were not significantly different than the control group (β = 0.41, t = 0.56, p > 0.05). The adjusted R² value of this model was 0.598.

The results of these analyses indicate that there was limited evidence of the self-

	Coef.	SE	t-stat.	p-value	Sig
Intercept	-1.678	1.264	-1.327	0.185	
Self-Efficacy Before	-0.147	0.156	-0.941	0.347	
Project Scores Before	-0.014	0.034	-0.420	0.675	
Pride Before	0.136	0.158	0.861	0.389	
Inadequate Before	-0.329	0.178	-1.852	0.065	
Frustrated Before	0.498	0.178	2.806	0.005	**
Self-efficacy During	0.727	0.091	7.984	0.000	***
Project Scores During	0.017	0.008	2.105	0.036	*
Frustrated During	-0.076	0.104	-0.733	0.464	
Inadequate During	0.063	0.100	0.628	0.530	
Pride During	0.058	0.095	0.613	0.540	
Completed Self-Evaluation	0.408	0.729	0.561	0.575	

Table 4.58: Intervention Effects: Self-efficacy Scores after the Intervention

evaluation intervention providing a benefit to those students that took part. Only one of the four models above showed a statistically significant effect of completing the intervention, which showed that while students were doing the self-evaluation task, they got higher project scores. These results are limited in the extent to which they justify conclusions about causal effects of the self-evaluation task, as covariates not included in the model may account for the observed effect of the intervention. A true randomized trial of the self-evaluation task, where students are required to complete it, would provide better evidence in this respect.

CHAPTER 5

DISCUSSION

5.1 Discussion of results

This study examined the influence of cognitive, affective, and dispositional factors on students' learning outcomes in a programming course. Using path analysis, this study examined the interaction of personality traits, self-regulated learning characteristics, and problem-solving ability, and their influence on programming project scores and exam scores. The results of this study indicated that problem-solving ability and conscientiousness had the greatest positive influence on students' learning outcomes when learning to program. Specifically, problem-solving ability had the strongest direct effect on both outcomes, whereas conscientiousness had a significant positive direct effect on programming project scores, and extraversion had a significant negative direct effect on exam scores. Using multiple regression, this study also investigated which of the cognitive, affective, and dispositional factors were the strongest predictors of project and exam scores. Problem solving was again the strongest predictor of both project and exam outcomes, and this held true for both male and female students. Conscientiousness also significantly predicted project scores, aligning with the path analysis results. This study also examined the influence of affective responses to programming projects on students' outcomes on those projects, focusing on effects over time. The results of this analysis exhibited that although there was significant covariance between programming projects and students' emotional responses to those projects, self-efficacy was the only factor that showed any significant reciprocal influence on project performance. Finally, this study analyzed the impact of a self-evaluation intervention on students' project scores and self-efficacy, both during the intervention and after the intervention ceased. The results of this analysis suggested that the self-evaluation intervention had a significant impact on project scores while it was occurring, but this impact on project scores did not

continue after the intervention ended, whereas there was no impact of the intervention on self-efficacy.

One of the primary goals of this study was to develop a detailed picture of how problem-solving ability, self-regulated learning characteristics, and personality traits interact to impact students' programming success outcomes. Path analysis models suggested that problem solving and conscientiousness had the strongest direct effects on project outcomes, and extraversion had a significant negative effect on exam scores. The path analysis results also uncovered a number of significant intermediate relationships between some of the factors being examined. For example, problem-solving had the largest direct effect on both learning outcomes, but the biggest predictor of problem-solving was openness to experience. In addition, several variables had a significant direct effect on self-efficacy, including openness to experience, intrinsic goal orientation, and metacognitive self-regulation. Likewise, conscientiousness and openness to experience had a significant direct effect on metacognitive self-regulation, and openness to experience was also significantly related to intrinsic goal orientation. Furthermore, emotional stability was negatively related to extrinsic goal orientation, whereas extraversion was positively related to extrinsic goal orientation. There were also some interesting non-findings, such as the fact that none of the self-regulated learning variables were significantly related to students' outcomes. Many of these findings aligned with previous research, whereas some of the findings surprisingly did not support prior research.

The variables that had significant direct effects on students' learning outcomes were problem solving, conscientiousness, and extraversion. These findings align with previous research which has suggested that problem solving is related to programming course outcomes (e.g. Mayer et al., 1986), conscientiousness is significantly related to postsecondary academic outcomes (Kappe and Van Der Flier, 2012), and extraversion is negatively related to academic performance if at all (Connor and Paunonen, 2007). The existing research on learning to program has suggested that problem solving is a significant predictor of programming course outcomes (Hostetler, 1983; Mayer et al., 1986). However, these previous studies on problem solving were not consistent on how problem solving was defined and measured (Lishinski et al., 2016a), so this study addressed that issue by using a robust measurement model for the problem-solving assessment based on the PISA problem solving items. Likewise, previous research that has examined the connection between the Big Five personality traits and students' academic outcomes has identified conscientiousness as the strongest predictor (Kappe and Van Der Flier, 2012), particularly at the postsecondary level (Komarraju et al., 2009). Therefore, it is not surprising that, of the Big Five traits, conscientiousness was the only one significantly related to programming project outcomes. It was somewhat surprising, however, that conscientiousness was related to project scores and not exam scores. Given that conscientiousness measures an individual's ability to be organized, disciplined, and dependable, one possible explanation for this finding is that successfully completing a programming project has a significant element of diligence and planning that is not required to do well on a multiple-choice exam. This hypothesis is supported by a previous study which found that conscientiousness was positively related to fewer errors found in programming code (Gnambs, 2015). The finding that extraversion had a significant negative relationship with exam scores also lined up well with previous studies which have found correlations between extraversion and a greater number of errors found in programming code (Gnambs, 2015), as well as similar findings for exam results in introductory psychology (Busato et al., 2000) and statistics courses (Furnham and Chamorro-Premuzic, 2004). Previous research on the connection between extraversion and academic outcomes has found either that extraversion has significant negative relationships with academic outcomes, or no relationship (Connor and Paunonen, 2007). This research has hypothesized that this finding is explained by greater time spent studying for introverts versus greater time spent socializing for extroverts (Chamorro-Premuzic and Furnham, 2014), but future research should examine programming students to determine whether or not their level of extraversion is related to their study behaviors, and if so, look for ways to help more extraverted programming

students pursue good study behaviors.

The most surprising lack of a finding from the path analysis was that none of the self-regulated learning constructs (i.e. goal orientation, self-efficacy, and metacognitive self-regulation) had significant direct effects on students' learning outcomes. These results contradict previous research, in which goal orientation (Zingaro and Porter, 2016), selfefficacy (Watson et al., 2014), and metacognitive self-regulation (Bergin and Reilly, 2005) have been found to be significant predictors of success in CS courses. These SRL constructs have also been found to be significantly associated with academic outcomes across settings (Valentine et al., 2004; Wolters and Yu, 1996; Zimmerman, 1990). Prior work in this area has only examined these student factors in isolation and not within a comprehensive model that also included cognitive variables and personality traits. Therefore, one possible explanation for the lack of significant SRL effects in this study could be that including more student factors confounded the effects of the SRL variables.

Of the four SRL variables examined in this study, it was probably most surprising that there were no significant direct effects of self-efficacy on either project or exam scores. This lack of a finding contradicts the previous research which has overwhelmingly linked self-efficacy to academic success outcomes across fields (Valentine et al., 2004), as well as in computer science (Watson et al., 2014). One possible explanation for this is that other variables examined in this study confounded the influence of self-efficacy on the learning outcome variables, which is plausible given that there were several variables included in the path model that had a significant direct effect on self-efficacy (metacognitive self-regulation, intrinsic goal orientation, openness to experience, and extrinsic goal orientation). Another possible explanation is that self-efficacy is very time point specific, particularly for individuals having new experiences. Individuals' self-efficacy is dynamic and subject to revision as one has experiences of success or failure (Bandura, 1977). It is for this reason that Bandura (1977) suggested that self-efficacy be measured repeatedly at "significant junctures in the change process." Results related to the reciprocal effects of emotional responses on performance in this study found significant connections between self-efficacy (the "will do well" item) and project score outcomes over time. This finding lends support to the hypothesis that programming students change their self-efficacy beliefs frequently in response to experiences of success or failure during the course. So perhaps the lack of a connection between initial self-efficacy and course outcomes has to do with later self-efficacy beliefs being more salient as students revise their beliefs over time. Prior results have also shown that introductory programming students continually revise their self-efficacy beliefs and that later self-efficacy beliefs predict outcomes better than earlier ones when both are included in a model (Lishinski et al., 2016b). Another possible explanation for the lack of significant direct effects of selfefficacy is that self-efficacy was measured using a generic item rather than items specific to the programming endeavor. Self-efficacy is very task specific, which led Bandura (1977) to warn that correlations based on aggregate self-efficacy measures may not fully reveal the degree of correspondence between self-efficacy and performance if the efficacy beliefs do not correspond precisely to the outcome tasks being measured. Given the results of this study, one possible approach to future research on self-efficacy could be to use an instrument more tailored to specific programming tasks (e.g. Bhardwaj, 2017). Using such an instrument may reveal more connections between self-efficacy and programming outcomes.

The path analysis revealed some significant intermediate relationships, which have a basis in previous research. The biggest predictor of learning outcomes was problem solving ability, which the path model exhibited was predicted by openness to experience. This is consistent with prior research, which has found that higher openness is associated with better decision making in a set of problem solving tasks, particularly when the context of the task unexpectedly changed (Le Pine et al., 2000). This is not surprising given that greater openness is associated with greater adaptability, which is important for task performance in a novel, complex, or ill-defined environment (Le Pine et al., 2000; Mumford et al., 1993). The goal of the PISA problem solving assessment was to put students in ill-defined situations where the method of solution is not readily apparent and could not be ascertained based on previous knowledge (PISA, 2003), which could explain explains why openness was related to problem solving. Le Pine et al. (2000) also found that individuals with higher conscientiousness did worse at decision making in the problem solving task, which was related to their tendencies toward order, dutifulness, and deliberation. This finding could further help explain why conscientiousness was correlated with project scores but not exam scores, as programming projects require students to plan ahead and take the time to check their work, whereas the exam context does not reward conscientiousness in that way.

The observed predictors of self-efficacy in the path models (openness to experience, intrinsic goal orientation, and metacognitive self-regulation) have all been found to predict self-efficacy in previous research as well. Openness to experience has been found to predict self-efficacy, as individuals who are intellectually curious, creative, open to new and different experiences, and have a tolerance for ambiguity have been found to have greater confidence that they will succeed in unfamiliar experiences and settings (Judge et al., 1999; Thoms et al., 1996), including academic settings (Caprara et al., 2011). Given that programming is unlike most other academic experiences that students encounter and has a reputation for being difficult, it is not surprising that higher openness would give them more confidence. Previous research has also found that intrinsic goal orientations predict self-efficacy beliefs because a mastery orientation protects individuals from the negative effects of failure, namely lowered self-efficacy beliefs (Bell and Kozlowski, 2002). Given that programming has a steep learning curve, sometimes with early failure experiences and emotional difficulties (Kinnunen and Simon, 2011), it is no surprise that students who are more resistant to the negative effects of failure will be more likely to believe that they will succeed. Likewise, previous research has also found that metacognitive self-regulation and self-efficacy are related (Chen et al., 2004; Credé and Phillips, 2011).

Using regression analysis, this study also examined which factors were the most important predictors of student learning in programming. Whereas path analysis allowed for the complex interrelationships between the many variables to be modeled, multiple regression simply determined which of the explanatory variables, holding the others constant, explained the most variance in the outcome variables, which the path analysis would not necessarily show. Overall, the multiple regression results told a similar story to the path analysis results about the important student factors in introductory programming. Specifically, the results exhibited that problem solving was the most important factor when learning to program for both project and exam outcomes, whereas conscientiousness also predicted project outcomes. Given this convergence of the multiple regression and path analysis results, the biggest takeaway from this study is that that in order to succeed at programming, it is helpful to be skilled at solving problems generally. Prior work has mainly involved examining how programming influences problem solving ability (Palumbo, 1990), but there is research emerging which has suggested that problem solving is an important predictor of introductory programming student learning outcomes (Lishinski et al., 2016a). However, problem solving explained relatively little variance in outcomes in the multiple regression models. In both the project and exam score models, problem solving (along with all the other factors) explained less than 10% of the variance in outcomes. Once again, one possible explanation for this somewhat surprising outcome is simply that a more fine-grained look at students' programming experiences during the course and examination of how these experiences influence students over time may be necessary to better predict programming student outcomes.

This study also investigated students' emotional responses to programming projects, looking for evidence of reciprocal effects between project scores and emotional reactions. The more evidence of reciprocal effects that could be found (i.e., significant cross-lagged paths), the more evidence that there would be of a feedback loop process with students' emotional responses to projects. The autoregressive cross-lagged path models tested whether there were such reciprocal effects between each of the emotional responses (frustration, inadequacy, pride, and self-efficacy) and project scores by fitting a path model including cross-lagged paths between emotional responses and project scores at different time points to model reciprocal effects, and autoregressive paths to control for prior values of emotional responses and project scores.

The results suggested that only the self-efficacy model showed substantial evidence of reciprocal effects, as three of the four cross-lagged paths were significant. This pattern of results suggested that there were reciprocal effects, but that these perhaps tapered off towards the end of the semester. One possible explanation for this may be that students are less prone to revise their self-efficacy beliefs in a reactionary way as they gain more competence with troublesome threshold concepts, which have been found to be a significant source of emotional difficulty for students (Eckerdal et al., 2007). This is consistent with previous research, which has found that emotionally difficult experiences in programming drive students' revisions of their self-efficacy beliefs (Kinnunen and Simon, 2011).

The fact that the most evidence for reciprocal effects was found for self-efficacy conforms with previous research, which suggests that there should be reciprocal effects between experiences of success or failure and self-efficacy belief formation due to the nature of the construct (Bandura, 1977; Pajares, 1996; Zimmerman and Risemberg, 1997; Zimmerman, 1990). The limited empirical research on affective feedback loops has suggested that selfefficacy may have such a relationship with success outcomes (Caprara et al., 2008; Williams and Williams, 2010), as well as related self-evaluation constructs like perceived goal achievement (Waschle et al., 2014). Given this evidence from previous research for self-efficacy reciprocal effects, along with the evidence for reciprocal effects of self-efficacy in this study, further research could investigate whether different measurement approaches to self-efficacy show different levels of reciprocal effects. Specifically, researchers could use programmingspecific self-efficacy questions (e.g. Ramalingam and Wiedenbeck, 1999), assess self-efficacy in courses beyond CS1 (Danielsiek et al., 2017), and attempt to determine which junctures of the course would be most significant for assessing self-efficacy.

While the autoregressive and cross-lagged paths showed the relationship between emotional responses and project scores over a period of time, the estimated covariances showed this relationship at the same time. An interesting pattern of results with the covariances emerged, as there seemed to be different patterns for the positive and negative valence emotional reactions. The two negative emotional responses (frustration and inadequacy) showed a similar pattern in which the covariance between the emotional reaction and project scores was the highest at the beginning of the course, and diminished to be statistically insignificant by the end of the course. The two positive emotional responses (pride and self-efficacy) showed a similar pattern in which the covariance between the emotional response and project scores peaked in the middle of the course. One possible explanation for these patterns is that the negative emotions were the most salient at the beginning of the course because of initial difficulty with the steep learning curve of programming. Then in the middle of the course, positive emotions became salient because students began to achieve more and feel some mastery, but by the end of the semester, students had overcome the most difficult part of the learning curve and gained enough experience and proficiency that they did not have as strong of emotional responses (positive or negative) or concomitant revisions of their self-efficacy beliefs.

The significant covariances, but lack of significant cross-lagged effects, in the frustration, inadequacy, and pride models means that these emotional reactions were significantly associated with the outcomes of projects during the same time chunk, but effects on later projects were not significant. This suggests that perhaps, while the projects can cause strong emotional reactions, these reactions tend to dissipate without creating any sustained effects. This is a positive finding for students and teachers, because it seems that successive mastery of the content is the most influential factor, rather than the emotions students experienced during the learning. Overall, the results of these analyses revealed that while students may have had strong emotional reactions to programming projects, they did not become mired in feedback loops where emotional reactions impeded their future performance.

This study also examined the impact of an intervention to bolster students' selfefficacy, and in turn their project score outcomes. The intervention gave students a selfevaluation task in order to provide them with opportunities to engage in self-reflection. The intervention was based on previous research which had tested a similar self-evaluation intervention as a way of bolstering self-efficacy (Schunk and Ertmer, 2000), as well as the theoretical framework of self-regulated learning, which suggests that good metacognitive self-regulators should engage in tasks such as self-evaluation and self-reflection (Zimmerman, 1986, 2002). The hypothesis was that the intervention would induce students to engage in these behaviors, which previous research, as well as the results of the path analysis in this study, have shown are significantly related to self-efficacy (Credé and Phillips, 2011).

Overall, the results of the regression analysis suggested that the intervention did not have significant effects on self-efficacy, or on project scores after the intervention was over. The results did, however, suggest a significant effect of the intervention on project scores during the intervention weeks. These results suggest that perhaps the self-evaluation task positively influenced project performance, but only while students were actively engaged in it and it did not have a lasting influence on student performance. Future research could investigate whether instructors could provide opportunities for students to engage in selfevaluation of their work across the whole semester to positively impact their learning.

One possible explanation for why the intervention did not influence students' selfefficacy could be that the intervention was placed at the middle of the semester. Self-efficacy theory has suggested that self-efficacy beliefs are the most malleable at the beginning of learning something new, because the beliefs are built on experiences of success and failure, and so at the beginning of learning something new each experience would have more weight than it would if it occurred later (Bandura, 1977). This dovetails with the research on programming students revising their self-efficacy beliefs, which has identified the beginning of the learning process as the time when students are most vulnerable to the negative effects on their self-efficacy of repeated failures (Kinnunen and Simon, 2011). This explanation also fits with the results of this study which showed that self-efficacy had reciprocal relationships with outcomes at the beginning of the course that tapered off and students were less likely to use the performance feedback received from the projects to substantially revise their self-efficacy beliefs as the course went on. So future research could examine whether this intervention or a different self-efficacy bolstering intervention might have a greater impact on students' self-efficacy at the beginning of a course.

5.2 Implications

Although the findings of this study did not identify student factors that had very strong associations with students' learning outcomes, there are nevertheless some takeaways for CS education researchers and practitioners. One of the takeaways is that problem solving ability was a significant predictor of programming course success, which could be important for understanding the cognitive dimension of programming. In practical terms, the results of this study show that, controlling for all of the other factors considered in this study, an average student would gain about 3-5% in their course outcomes by being one standard deviation better at problem solving. Future research could develop a more fine-grained understanding of aspects of problem solving that are relevant to programming, such as abstraction. Researchers could also study how interventions could be used to bolster students' problem solving skills in order to increase their success in programming. Developing a problem-solving intervention that could increase the chances of success for introductory programming students could potentially be very important for broadening CS participation. Practitioners could also potentially make use of the findings about problem solving to identify students who are more likely to have difficulties in a programming course and provide them with additional support.

The findings on students' emotional reactions also have some takeaways for researchers and practitioners. Results from the self-efficacy model showed that self-efficacy was dynamic and that it had indirect reciprocal effects on students' performance in programming. However, this study measured students' self-efficacy by a single generic item; future research could further investigate the connection between self-efficacy and outcomes in programming courses by using multiple items that are tailored to the specific skills being developed in the course. The intervention used in this study, or some other sort of self-efficacy bolstering intervention, also provides a foundation for further research in programming courses. Future research could modify the self-evaluation intervention and/or conduct it as a true randomized experiment to investigate any effects on learning outcomes. Research has offered a number of suggestions for ways that teachers could improve students' self-efficacy, including moderating the difficulty of tasks, teaching specific learning strategies, reinforcing students based on effort and strategy use, stressing effort in their instruction, and by giving frequent and focused feedback (Kinnunen and Simon, 2011; Margolis and McCabe, 2006). Future research in CS education should develop and test pedagogical interventions using one or more of these strategies to determine how to best bolster students' self-efficacy when learning to program. Practitioners should take from this study an understanding of how self-efficacy influences students' outcomes as they progress through the learning process. Likewise, practitioners could use the self-evaluation task or a similar task in their own courses, with the goal of encouraging self-reflective behaviors in order to boost students' performance on course assignments.

5.3 Limitations

This study has a few main limitations regarding the generalizability of the results. As is always the case, the scope of data collection was limited, so only a limited number of factors could be examined, which makes it impossible to know for sure that the best ones were included. As detailed in the literature review section, the factors that were included were chosen because they had good potential for an impact on student outcomes in CS based on previous research. However, there could be other cognitive, affective, and dispositional characteristics that impact student outcomes that were not included in this study. Therefore, it's impossible to rule out the possibility of other variables that would have done a better job of explaining learning outcomes in this study. Another limitation to the generalizability of the study is that data came from a single context. The data in this study came from one introductory CS course at large Midwestern university with a single main instructor (although individual sections were lead by several different TAs). This limits the generalizability of the study results to the extent that the student population and organization of the course differ in other CS1 contexts at other universities and with other instructors.

Another limitation was that the validity of the conclusions of this study was limited by the measurement validity of the surveys and assessments. This is true of any study using surveys, but there is one way in particular that this study was limited by the instruments. As previously discussed with regard to SRL constructs, the results of this study may have been limited by the use of generic instruments rather than ones tailored to the CS context. It may have benefitted this study to be able to assess SRL constructs using measures that are specific to the context of a programming course. Although they do not currently exist, it is easy to imagine developing parallel CS specific measures for other SRL constructs like goal orientation. The relationships between SRL and programming course outcomes may have been occluded by not using CS specific measures.

Lastly, the results of this study have limited value for drawing causal conclusions. With the exception of the self-evaluation intervention, this was a purely correlational study, which has many well-known limitations for making broad and generalizable causal claims. Finding a significant relationship between two variables through structural equation modeling or multiple regression analysis does not provide the means for making causal inferences. For this reason, researchers should be cautious in drawing causal inferences about any of the constructs, but rather, they should interpret the results appropriately in light of the inherent limitations of the statistical methods that underlie them. Future research could use these results as a basis for knowing which constructs merit further attention, and then conducting experimental studies that would enable causal conclusions to be drawn.

The data from the self-evaluation is also limited for drawing causal conclusions

due to self-selection bias in the intervention. Furthermore, there were significant preexisting differences observed between the intervention and the non-intervention group on project scores, self-efficacy, and pride, suggesting that there was selection bias. The four outcome models that were used to analyze the impact of the intervention for the different outcomes controlled for the prior values of these variables. In the three cases where no significant difference between the groups was observed in the outcome model (the two self-efficacy outcomes and project scores after the intervention), these covariates (prior values of project scores, self-efficacy, and emotional responses) were sufficient to control for prior differences between the groups. Nevertheless, in the case of project scores during the intervention, where there was a significant difference between the groups in the outcome model, it remains a possibility that other unobserved preexisting differences between the groups could explain the difference attributed to the intervention. Future work could use a randomized experiment to address these concerns or a quasi-experimental method such as propensity score analysis to match participants on covariates prior to the intervention. Thus, the results on the influence of the self-evaluation intervention should be understood in light of the limitations created by the possibility of self-selection bias, which severely undermines any inference of causality about the effect of the intervention.
APPENDICES

APPENDIX A

MSLQ ITEMS

For each of the following statements, please choose how true each statement is of you.

7 point Likert scale:
1 = Not true of me at all
3
4 = Somewhat true of me
5
6
7 = Extremely true of me

A.1 Metacognitive Self-regulation

- 1. During class time I often miss important points because I'm thinking of other things.
- 2. When reading for this course, I make up questions to help focus my reading.
- 3. When I become confused about something I'm reading for this class, I go back and try to figure it out.
- 4. If course materials are difficult to understand, I change the way I read the material.
- 5. Before I study new course material thoroughly, I often skim it to see how it is organized.
- 6. I ask myself questions to make sure I understand the material I have been studying in this class.
- 7. I try to change the way I study in order to fit the course requirements and instructor's teaching style.
- 8. I often find that I have been reading for class but don't know what it was all about.
- 9. I try to think through a topic and decide what I am supposed to learn from it rather than just reading it over when studying.
- 10. When studying, I try to determine which concepts I don't understand well.
- 11. When I study, I set goals for myself in order to direct my activities in each study period.
- 12. If I get confused taking notes in class, I make sure I sort it out afterwards.

A.2 Self-efficacy

- 1. I believe I will receive an excellent grade in this class.
- 2. I'm certain I can understand the most difficult material presented in the readings for this course.
- 3. I'm confident I can understand the basic concepts taught in this course.
- 4. I'm confident I can understand the most complex material presented by the instructor in this course.
- 5. I'm confident I can do an excellent job on the assignments and tests in this course.
- 6. I expect to do well in this class.
- 7. I'm certain I can master the skills being taught in this class.
- 8. Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this class.

A.3 Intrinsic Goal Orientation

- 1. In a class like this, I prefer course material that really challenges me so I can learn new things.
- 2. In a class like this, I prefer course material that arouses my curiosity, even if it is difficult to learn.
- 3. The most satisfying thing for me in this course is trying to understand the content as thoroughly as possible.
- 4. When I have the opportunity in this class, I choose course assignments that I can learn from even if they don't guarantee a good grade.

A.4 Extrinsic Goal Orientation

- 1. Getting a good grade in this class is the most satisfying thing for me right now.
- 2. The most important thing for me right now is improving my overall grade point average, so my main concern in this class is getting a good grade.

- 3. If I can, I want to get better grades in this class than most of the other students.
- 4. I want to do well in this class because it is important to show my ability to my family, friends, employer, or others.

APPENDIX B

BIG 5 ITEMS

For each of the following statements, please choose how true each statement is of you.

7 point Likert scale:
1 = Not true of me at all
3
4 = Somewhat true of me
5
6
7 = Extremely true of me

B.1 Extraversion

- 1. I Am the life of the party.
- 2. I Feel comfortable around people.
- 3. I Start conversations.
- 4. I Talk to a lot of different people at parties.
- 5. I Don't mind being the center of attention.
- 6. I Don't talk a lot.
- 7. I Keep in the background.
- 8. I Have little to say.
- 9. I Don't like to draw attention to myself.
- 10. I Am quiet around strangers.

B.2 Agreeableness

- 1. I Am interested in people.
- 2. I Sympathize with others' feelings.

- 3. I Have a soft heart.
- 4. I Take time out for others.
- 5. I Feel others' emotions.
- 6. I Make people feel at ease.
- 7. I Am not really interested in others.
- 8. I Insult people.
- 9. I Am not interested in other people's problems.
- 10. I Feel little concern for others.

B.3 Conscientiousness

- 1. I Am always prepared.
- 2. I Pay attention to details.
- 3. I Get chores done right away.
- 4. I Like order.
- 5. I Follow a schedule.
- 6. I Am exacting in my work.
- 7. I Leave my belongings around.
- 8. I Make a mess of things.
- 9. I Often forget to put things back in their proper place.
- 10. I Shirk my duties.

B.4 Emotional Stability

1. I Am relaxed most of the time.

- 2. I Seldom feel blue.
- 3. I Get stressed out easily.
- 4. I Worry about things.
- 5. I Am easily disturbed.
- 6. I Get upset easily.
- 7. I Change my mood a lot.
- 8. I Have frequent mood swings.
- 9. I Get irritated easily.
- 10. I Often feel blue.

B.5 Openness to Experience

- 1. I Have a rich vocabulary.
- 2. I Have a vivid imagination.
- 3. I Have excellent ideas.
- 4. I Am quick to understand things.
- 5. I Use difficult words.
- 6. I Spend time reflecting on things.
- 7. I Am full of ideas.
- 8. I Have difficulty understanding abstract ideas.
- 9. I Am not interested in abstract ideas.
- 10. I Do not have a good imagination.

APPENDIX C

SELF-EVALUATION SURVEY

Reflecting on the feedback you've just received for your project, and your code that you turned in, please respond to the following questions:

- What are 3 aspects of the work that you did in your project that you feel satisfied with, and confident about going forward?
- What are 3 aspects of the work that you did in your project that you DON'T feel satisfied with or confident about going forward?

APPENDIX D

PROBLEM SOLVING ITEMS

D.1 Library System

The John Hobson High School library has a simple system for lending books: for staff members the loan period is 28 days, and for students the loan period is 7 days. The following is a decision tree diagram showing this simple system:



The Greenwood High School library has a similar, but more complicated, lending system:

- All publications classified as "Reserved" have a loan period of 2 days.
- For books (not including magazines) that are not on the reserved list, the loan period is 28 days for staff, and 14 days for students.
- For magazines that are not on the reserved list, the loan period is 7 days for everyone.

• Persons with any overdue items are not allowed to borrow anything.

Q1: You are a student at Greenwood High School, and you do not have any overdue items from the library. You want to borrow a book that is not on the reserved list. How long can you borrow the book for?

Q2: Develop a decision tree for the Greenwood High School Library system so that an automated checking system can be designed to deal with book and magazine loans at the library. Your checking system should be as efficient as possible (i.e. it should have the least number of checking steps). Note that each checking step should have only two outcomes (e.g. "Yes" and "No"). Enter your decision tree in the blanks below by entering the check step questions, and what result follows if the answer is yes. If the answer is no, the decision tree proceeds to the next check step question.

For example, in the above chart, the answers would be as follows:

- Check step 1 question: Is the borrower a staff member?
- If Yes: Loan period is 28 days
- If No: Loan period is 7 days.

Blanks:

- Check step 1 question
- If Yes:
- Check step 2 question
- If Yes:
- Check Step 3 question
- If Yes
- Check Step 4 question
- If Yes:
- If No:

D.2 Trip

This problem is about planning the best route for a trip. Figures 1 and 2 show a map of the area and the distances between towns.



Figure 1: Map of roads between towns.



Angaz						
Kado	550					
Lapat	500	300				
Megal	300	850	550			
Nuben	500		1000	450		
Piras	300	850	800	600	250	

Q1: Calculate the shortest distance by road between Nuben and Kado Q2: Zoe lives in Angaz. She wants to visit Kado and Lapat. She can only travel up to 300 kilometres in any one day, but can break her journey by camping overnight anywhere between towns.

Zoe will stay for two nights in each town, so that she can spend one whole day sightseeing in each town.

Show Zoe's itinerary by completing determining where she stays each night.

Day 1: Camp sight between Angaz and Kado

Day 7: Angaz

Fill in days 2-6

D.3 Irrigation

Below is a diagram of a system of irrigation channels for watering sections of crops. The gates A to H can be opened and closed to let the water go where it is needed. When a gate is closed no water can pass through it. This is a problem about finding a gate which is stuck closed, preventing water from flowing through the system of challenges.



Michael notices that the water is not always going where it is supposed to. He thinks that one of the gates is stuck closed, so that when it is stuck to "open", it does not open. Q1: Michael uses the settings in table 1 to test the gates.



With the gate settings as given in table 1, list all of the possible paths for the flow of water (Example format: ABCD, use a new line for each different paths). Assume that all gates are working according to the settings.

Q2: Michael finds that, when the gates have the table 1 settings, no water flows through, indicating that at least one of the gates set to "open" is stuck closed. Decide for each problem case below whether the water will flow through all the way. Select yes or no in each case, Will the water flow through all the way?

- Gate A is stuck closed. All other gates are working properly as indicated in table 1.
- Gate D is stuck closed. All other gates are working properly as indicated in table 1.
- Gate F is stuck closed. All other gates are working properly as indicated in table 1. Q3: Michael wants to be able to test whether gate D is stuck closed.

Select settings (open/closed) below for the gates (A - G) to test whether gate D is stuck closed when it is set to "open."

D.4 Design by Numbers



Q1: Which of the following commands generated the graphic shown below?



- Paper 0
- Paper 20
- Paper 50
- Paper 75

Q2: Which of the following sets of commands generated the graphic shown below?



- Paper 100 Pen 0 Line 80 20 80 60
- Paper 0 Pen 100 Line 80 20 60 80
- Paper 100 Pen 0 Line 20 80 80 60
- Paper 0 Pen 100 Line 20 80 80 60

Q3: The following shows an example of a "repeat" command. Paper 0 Pen 100 Repeat A 50 80

Line 20 A 40 A

The command "Repeat A 50 80" tells the program to repeat the actions in brackets { }, for successive values of A from A=50 to A=80.



Write commands to generate the following graphic:



APPENDIX E

EMOTIONAL RESPONSE QUESTIONS

When you have completed the project insert the 5-line comment specified below. For each of the following statements, please respond with how much they apply to your experience completing the programming project, on the following scale:

1 = Strongly disagree / Not true of me at all
3
4 = Neither agree nor disagree / Somewhat true of me
6
7 = Strongly agree / Extremely true of me
Q1: Upon completing the project, I felt proud/accomplished
Q2: While working on the project, I often felt frustrated/annoyed

Q3: While working on the project, I felt inadequate/stupid

Q4: Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this course.

APPENDIX F

EXAM EXAMPLE

01. If x = 23 and y = 5, what is x//y? A) 5 B) 4.6 C) 4.0 D) 4 E) None of the above. 02. What is printed by print(2+2**3) A) 8 B) 10 C) 36 D) 64 E) None of the above. 03. What is the value returned by 9 % 4 ? A) 0 B) 1 C) 2 D) 2.25 E) None of the above. 04. Given a = True, b = False What is printed by print(a or b and b) A) 0 B) 1 C) True D) False E) None of the above. 05. What is the decimal value of the binary number 110? A) 110 B) 7 C) 6

```
D) 4
     E) None of the above.
06. Given a = 4; what is printed by \# careful with this one
        if a == 4 and 5:
            print("It is True!")
        else:
            print("No!")
     A) It is True!
     B) It is False!
     C) No!
     D) None
     E) None of the above.
07. Given S = "Go Green!"
    What is returned by S[:3]?
    A) 'Go'
     B) 'Go '
     C) 'en!'
     D) 'n!'
     E) None of the above.
08. Given S = "Go Green!"
    What is returned by S[4:-1]?
     A) '!neerG oG'
     B) 'Green'
     C) 'Green!'
     D) 'reen'
     E) None of the above.
09. What is printed by the following?
   s = "dog"
   for i in s:
       print(i, end = '') # there is no space between the quotes
   A) 0 1 2
   B) d o g
   C) dog
  D) i i i
   E) None of the above.
```

```
10. If s = 'beat'
    how do I make s = 'beet' ?
     A) s = s[:2] + 'e' + s[-1]
     B) s = s[:2] + 'e' + s[-1:]
     C) s = s.replace('a', 'e')
     D) All of the above
     E) None of the above.
11. What is printed?
       s = '' # there is no space between the quotes
       if s:
          print("True")
       else:
          print("False")
     A) True
     B) False
     C) None
     D) None of the above.
12. Given ch = 'E'
    what is returned by ch < 'H'?
    A) True
    B) False
    C) None
    D) None of the above.
################
# Figure 1 #
HEX = '0123456789abcdefABCDEF'
  the_str = "My cell is 353-3389."
  h,i,j,k = 0,0,0,0 # initialize to zero
  for c in the_str:
      if c in HEX:
         h += 1
      elif c.isdigit():
          i += 1
      elif c.isalpha():
          j += 1
      else:
```

k += 1 print(h) # Line 1 print(i) # Line 2 print(j) # Line 3 # Line 4 print(k) 13. When Figure 1 is executed what is printed by Line 1? A) 7 B) 6 C) 5 D) 0 E) None of the above 14. When Figure 1 is executed what is printed by Line 2? A) 7 B) 6 C) 5 D) 0 E) None of the above 15. When Figure 1 is executed what is printed by Line 3? A) 7 B) 6 C) 5 D) 0 E) None of the above 16. When Figure 1 is executed what is printed by Line 4? A) 7 B) 6 C) 5 D) 0 E) None of the above ############ # Figure 2 # ############ try: fp = open("someFile")

```
except XXXX:
    print("Error")
```

17. Consider Figure 2. What is an appropriate replacement for XXXX?

- A) IOError
- B) FileNotFoundError
- C) FileError
- D) OpenFileError
- E) None of the above.

18. Which of the following opens the file "someFile" for reading?

```
A) open("someFile")
     B) open("someFile","r")
     C) open("someFile","read")
     D) All of the above.
     E) None of the above.
############
# Figure 3 #
############
  1345.68
  2345.68
  3345.68
  4345.68
  5345.68
12345678901
19. Given S = '345678'
    which code prints the first five lines in Figure 3?
    The digits 12345678901 are there to help you count spaces;
    they are not part of the print loop.
    Note the spacing and right-left justification.
    A) for i,j in enumerate(S):
           if i==0:
               continue
           print("{:>3d}{:6.2f}".format(i,float(j)))
    B) for i in range(1,len(S)):
           n = int(S)/1000
           print("{:>3d}{:6.2f}".format(i,n))
```

```
C) for i in range(1,len(S)):
           n = int(S)/1000
           print("{:3d}{:8.2f}".format(i,n))
    D) for i in range(1,len(S)):
           print("{:>3d}{:6f}".format(i,float(S)))
    E) None of the above.
20. Which code prints these numbers: 2 4 6 8
    A) for i in range(2,10,2):
           print(i, end = ' ')
    B) for i in range(4):
           print(2*i, end = ' ')
    C) for i in range(2,8):
           print(i, end = ' ')
    D) for i in range(1,4):
           print(2*i, end = ' ')
    E) None of the above.
 #############
 # Figure 4 #
 ############
  X = int(input("Input an integer: "))
  for i in range(5):
      if i%2 == 0:
          continue
      print(i, end = " ")
      if i == X:
          break
  else:
      print("XXXX") # Line 1
21. In Figure 4 what value of X causes Line 1 to be printed?
    A) 2
    B) 3
```

- C) 4
- D) 5

E) None of the above. 22. In Figure 4 what value of X causes Line 1 to NOT be printed? A) 2 B) 3 C) 4 D) 5 E) None of the above. ############ # Figure 5 # ############ def f1(a): a += 1 return a a = 1 print(f1(a)) # Line 1 print(a) # Line 2 a = f1(a)print(a) # Line 3 23. What is printed by Line 1 in Figure 5? A) 1 B) 2 C) 3 D) None E) None of the above. 24. What is printed by Line 2 in Figure 5? A) 1 B) 2 C) 3 D) None E) None of the above. 25. What is printed by Line 3 in Figure 5? A) 1 B) 2 C) 3

D) None

E) None of the above.

APPENDIX G

PROGRAMMING PROJECT EXAMPLE

Programming Project #6: Password File Cracker

(Edits: changed itertools permutations to product-either works for these passwords, but product is the correct one. Removed lists and tuples from the forbidden list.)

This project is worth 45 points (4.5% of the courses grade). It uses error checking, functions, and opening files and must be completed and turned in before 11:59 PM on Monday, February 27th.

Assignment Background

Over the past several decades computer systems have relied on the concept of a password in order to authenticate users and grant access to secure systems. I hope you all are using several different passwords for different systems and websites as that is currently the best security practice. However even when requiring users to use "strong passwords" and change them frequently, passwords inevitably get leaked and people unfortunately use the same password for their bank account as for Facebook.

There are a vast array of methods for cracking passwords but this project will introduce you to 2 of them. Brute force and dictionary.

A brute force attack is a computer program will generate every possible combination of input then try all possibilities. Take a smartphone as an example, most have a 4 digit pass code so a brute force attack would start by first trying 0000, 0001, 0002, 0003, 0004, so on and so forth, until the correct passcode is found. In the case of a 4-digit password, there are 10,000 combinations. When using all the characters on your keyboard, the possible combinations quickly climb into the millions.

Given enough time and computing power, a brute force attack will always work. However, keep in mind though that most modern encryption standards would take several thousand years and require a lot of computing power to generate all possible combinations. So, if you're encrypting your devices (as you should be) and using strong passwords you're probably safe from a brute force attack. A dictionary attack relies on a source file (like the one we provide you) in order to carry out an attack. A source file is simply just a text file containing passwords. Sometimes, when malicious hackers manage to break into a "secure" system they release a password dump which is a file containing all the passwords found on that system. This is a good way of seeing the most common passwords.

Note: there are several ways of cracking / stealing passwords that can range from tricking people into telling you (known as phishing) to breaking some vulnerability in a website (such as an SQL injection) and many more.

Assignment Specifications

You will be developing a python program that implements the two password cracking methods and allows the user to crack .zip (archive) files.

The first task is to implement a driver loop that prompts the user for which file cracking method they'd like to use (brute force, dictionary, or both). After determining which method the user would like to use, call the respective function(s) and show the time each function took to run.

The user should be able to quit by typing "q". If the attack is 'both', do a dictionary attack first and do a brute force attack only if the dictionary attack failed.

Your second task is to develop a brute force function that generates a strings containing a-z (all lowercase) of length 8 or less as the password to attempt (Hint: start small trying small passwords first). For example ,your string should start out with "a", then "b", then "c", so on and so forth. When you get to "z" change the first character back to "a" and append another letter so your string becomes "aa" then "ab". Once the password is cracked you should display the password to the user. Hint: see notes below about itertools.

The third task is to then develop the dictionary function. After you have prompted for the name of the dictionary file and the target .zip file your function should go through every line (one password per line) and try opening the .zip file. If it opens successfully, then again, display the password to the user. If you try all the passwords in the source file, display an appropriate error message. Hint: remember to strip() the word read from the file before trying it as a password!

The fourth and final task for this project is a research question. With this project we are providing you with two password protected .zip files. On behalf of the Computer Science Department you have our full permission to attack these two specific files for demonstration and educational purposes only.

Breaking into secure computer systems is illegal in the US. (most other countries have similar laws). Your research question is to find out what US federal law you're breaking if you decide to do this without permission. Your program should warn the user what the name of the law is, and what the maximum penalty is before prompting for input.

Assignment Notes You are also not allowed to use sets, or dictionaries.

Divide-and-conquer is an important problem solving technique. In fact, inside of every challenging problem is a simpler problem trying to get out. Your challenge is to find those simpler problems and divide the problem into smaller pieces that you can conquer. Functions can assist in this approach. Hint: build the following functions one at a time and test each one before moving on to the next.

1. open_dict_file() function does not take in any arguments and should return a file pointer to the dictionary file. It should keep re-prompting the user if the file doesn't exist.

2. open_zip_file() function does not take in any arguments and should return a file pointer to the zip file. It should keep re-prompting the user if the file doesn't exist.

3. brute_force_attack(zip_file) function takes the zip file pointer as an argument. The brute force function will generate all the possibilities, and attempt them. It should output the correct password once found. If successful, return True, otherwise return False.

4. dictionary_attack(zip_file, dict_file) function takes the zip file pointer and dictionary file pointer as arguments. It should output the correct password once found or print an error message if not successful. If successful, return True, otherwise return False.

5. File reading and closing. The order of reading files is important for testing. When doing a 'dictionary' or 'both' attack,

prompt for the dictionary file before prompting for zip file(s). Also, always close() all files after each crack; otherwise with the dictionary file you will start reading passwords where you left off rather than at the beginning. 6. Using the code we provide for you, display the time need to finish each function. This will allow the user to see the time needed to complete various attacks. (You should notice quite a difference between the two methods). Deliverables The deliverable for this assignment is the following file: proj06.py -- your source code solution Be sure to use the specified file name and to submit it for grading via the handin system before the project deadline Working with Archive (zip) Files To properly open zip files in python you need to import the zipfile module. To attempt extracting a password protected zip file, use zip file = zipfile.ZipFile(filename) to first open the zip file, then apply your password to the opened zip file using zip_file.extractall(pwd=password.encode()) so you can extract all its members. The type of the variable password is string and encode() is a string method that you must apply to the password variable when used as an argument to extractall. Basically, you try to unzip the file using extractall: if it generates an error, the password didn't work; if it doesn't generate an error, the password succeeded. Therefore, you want to put the call to extractall within try-except. Unfortunately, a variety of errors get generated by extractall so you must use except with no error type specified (that is generally not good practice, but necessary in this case). Therefore, your except line is simply except: We provided two password protected zip files. One that can be cracked with the brute force method and one that can be cracked with the dictionary method.

Optional: If you would like to test your program with more than just our two zip files, here are instructions for creating a password protected zip file on Mac and Windows: Mac: http://osxdaily.com/2012/01/07/set-zip-password-mac-os-x/

Windows: (we recommend using the WinRar part of the tutorial) http://www.intowindows.com/how-to-create-zip-file-with-password-in-win dows-78/

Note: when password protecting your own zip file, make sure your password is a-z, lowercase, and less than 8 characters long. You're welcome to try longer passwords but the time required to break them will increase exponentially.

Working with itertools

Python has many, many useful modules and itertools is one that is useful in general and a major effort-savor for this project. To brute force passwords you want to try all permutations of characters. The itertools has a function named product that provides exactly that functionality. However, that function generates tuples of characters, e.g. ('a','b','c'), whereas we need a string, e.g. 'abc'. There is a string method named join that allows you to join characters together with a specified character between each one. For example, '-'.join(('a','b','c'))yields the string 'a-b-c', but we don't want any characters in between so we use an empty string as ''.join(('a','b','c'))to yield 'abc'. Experiment with these in the shell. For example, if we want to generate strings (passwords in this project) of length 3 from a

from itertools import product

for items in product('abcdef',repeat=3): print(''.join(items))

string of characters 'abcdef' we use the following:

Working with time

function took.

To determine time we use yet another module: time. To calculate the time it takes a function to run get the time before calling the function, call time again right after calling the function, and then take the difference and print. Use the time.process_time() function, e.g. to time the dictionary_attack function: start = time.process_time() dictionary_attack(zip_file, dict_file) end = time.process_time() Then take the difference, round to 4 decimal places and print. The time is in seconds and the individual values have no meaning other than that when you take the difference you get the time that the Grading Rubric Computer Project #06 Scoring Summary General Requirements 0 (5 pts) Coding Standard 1-9 (descriptive comments, function header, etc...) Implementation: 0 (10 pts) Pass test1: able to crack a zip file with a dictionary attack and no error in input 0 (10 pts) Pass test2: able to crack a zip file with a brute force attack and no error in input 0 (5 pts) Pass test3 and test4: Control and input: loops as specified and handles errors in input 0 (5 pts) Pass test5: 'both' selection with failed and successful dictionary attacks O (5 pts) Script contains warning explaining the US federal law on hacking 0 (5 pts) Cracking times are calculated and posted in seconds TA Comments: Sample Output (Note that all passwords are marked with X's - you are to figure them out. Also, the law and prison items are marked with X's for the same reason.) Test Case 1 Cracking zip files. Warning cracking passwords is illegal due to law XXXXX and has a prison term of XXXXX

What type of cracking ('brute force', 'dictionary', 'both', 'q'): dictionary Dictionary Cracking Enter dictionary file name: rockyou.txt Enter zip file name: dictionary attack.zip Dictionary password is XXXXXX Elapsed time (sec): 0.2016 What type of cracking ('brute force', 'dictionary', 'both', 'q'): q Test Case 2 Cracking zip files. Warning cracking passwords is illegal due to law XXXXX and has a prison term of XXXXX What type of cracking ('brute force', 'dictionary', 'both', 'q'): brute force Brute Force Cracking Enter zip file name: brute_force.zip Brute force password is XXXXXX Elapsed time (sec): 3.0039 What type of cracking ('brute force', 'dictionary', 'both', 'q'): q Test Case 3 Cracking zip files. Warning cracking passwords is illegal due to law XXXXX and has a prison term of XXXXX What type of cracking ('brute force', 'dictionary', 'both', 'q'): dictionary Dictionary Cracking Enter dictionary file name: xxx Enter dictionary file name: rockyou.txt Enter zip file name: xxxx Enter zip file name: yyyy file name: dictionary_attack.zip Dictionary password Enter zip is XXXXXX Elapsed time (sec): 0.1957 What type of cracking ('brute force', 'dictionary', 'both', 'q'): q Test Case 4

Cracking zip files. Warning cracking passwords is illegal due to law XXXXX and has a prison term of XXXXX What type of cracking ('brute force', 'dictionary', 'both', 'q'): q Test Case 5 (In Test Case 5 use the short dictionary and the brute force.zip file so the dictionary attack fails and doesn't take forever to do so-forcing the brute force attack to be needed.) Cracking zip files. Warning cracking passwords is illegal due to law XXXXX and has a prison term of XXXXX What type of cracking ('brute force', 'dictionary', 'both', 'q'): both Both Brute Force and Dictionary attack. Enter dictionary file name: short_dict.txt Enter zip file name: brute force.zip No password found. Dictionary Elapsed time (sec): 0.0188 Brute force password is XXXXXX Brute Force Elapsed time (sec): 2.9494 What type of cracking ('brute force', 'dictionary', 'both', 'q'): both Both Brute Force and Dictionary attack. Enter dictionary file name: rockyou.txt Enter zip file name: dictionary attack.zip Dictionary password is XXXXXX Dictionary Elapsed time (sec): 0.2314 What type of cracking ('brute force', 'dictionary', 'both', 'q'): q Optional Testing This test suite five tests for the entire program. 1. Testing the entire program using run_file.py. You need to have the files test1.txt to test5.txt from the project directory. Make sure that you have the following lines at the top of your program (only for testing): import sys def input(prompt=None): if prompt != None: print(prompt, end="") aaa_str = sys.stdin.readline() aaa_str =

aaa_str.rstrip("\n") print(aaa_str)
return aaa_str

BIBLIOGRAPHY

BIBLIOGRAPHY

- Ackerman, P. L. (1988). Determinants of Individual Differences During Skill Acquisition: Cognitive Abilities and Information Processing. *Journal of Experimental Psychology: Gen*eral, 117(3):288–318.
- Al-Harthy, I. S. and Was, C. A. (2013). Knowledge monitoring, goal orientations, self-efficacy, and academic performance: A path analysis. *Journal of College Teaching and Learning*, 10(4):263–279.
- Andrew, S. (1998). Self-efficacy as a predictor of academic performance in science. *Journal* of Advanced Nursing, 27(3):596–603.
- Ashcraft, M. H. and Krause, J. A. (2007). Working memory, math performance, and math anxiety. *Psychonomic Bulletin & Review*, 14(2):243–248.
- Bandura, a. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2):191–215.
- Bandura, A. (1986). Social Foundations of Thought and Action: A Social Cognitive Theory. Prentice-Hall, Englewood Cliffs, NJ.
- Bandura, A., editor (1995). Self-efficacy in Changing Societies. Cambridge University Press.
- Bandura, A. (1997). Self-efficacy: The Exercise of Control. Macmillan.
- Barker, R. J. and Unger, E. A. (1983). A predictor for success in an introductory programming class based upon abstract reasoning development. ACM SIGCSE Bulletin, 15(1):154–158.
- Bell, B. S. and Kozlowski, S. W. J. (2002). Goal orientation and ability: interactive effects on self-efficacy, performance, and knowledge. *The Journal of applied psychology*, 87(3):497– 505.
- Bennedsen, J. and Caspersen, M. E. (2005). An investigation of potential success factors for an introductory model-driven programming course. *Proceedings of the First International Workshop on Computing Education Research*, pages 155–163.
- Bennedsen, J. and Caspersen, M. E. (2007). Failure Rates in Introductory Programming. ACM SIGCSE Bulletin, 39(2):32–36.
- Bergin, S. and Reilly, R. (2005). Programming: Factors that Influence Success. ACM SIGCSE Bulletin, 37(1):411–415.
- Bergin, S., Reilly, R., and Traynor, D. (2005). Examining the role of self-regulated learning on introductory programming performance. *First International Workshop on Computing Education Research*, pages 81–86.

- Bernstein, D. R. (1991). Comfort and Experience with Computing: Are They the Same for Women and Men? *SIGCSE Bull.*, 23(3):57–61.
- Beyer, S. (2008). Predictors of Female and Male Computer Science Students' Grades. Journal of Women and Minorities in Science and Engineering, 14(4):377–409.
- Bhardwaj, J. (2017). In search of self-efficacy: development of a new instrument for first year Computer Science students. *Computer Science Education*, pages 1–21.
- Bidjerano, T. and Dai, D. Y. (2007). The relationship between the big-five model of personality and self-regulated learning strategies. *Learning and Individual Differences*, 17(1):69–81.
- Box, G. and Cox, D. (1964). An Analysis of Transformations. Journal of the Royal Statistical Society, 26(2):211–252.
- Britner, S. L. and Pajares, F. (2006). Sources of science self-efficacy beliefs of middle school students. *Journal of Research in Science Teaching*, 43(5):485–499.
- Broadbent, J. and Poon, W. L. (2015). Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *Internet and Higher Education*, 27:1–13.
- Brooks, R. (1999). Towards a Theory of the Cognitive Processes in Computer Programming. International Journal of Human-Computer Studies, 51(2):197–211.
- Browne, M. W. and Cudeck, R. (1993). Alternative ways of assessing model fit. Sage focus editions.
- Busato, V. V., Prins, F. J., Elshout, J. J., and Hamaker, C. (2000). Intellectual ability, learning style, personality, achievement motivation and academic success of psychology students in higher education. *Personality and Individual Differences*, 29(6):1057–1068.
- Busch, T. (1995). Gender Differences in Self-Efficacy and Attitudes Toward Computers. Journal of Educational Computing Research, 12(2):147–158.
- Caprara, G. V., Fida, R., Vecchione, M., Del Bove, G., Vecchio, G. M., Barbaranelli, C., and Bandura, A. (2008). Longitudinal analysis of the role of perceived self-efficacy for self-regulated learning in academic continuance and achievement. *Journal of Educational Psychology*, 100(3):525–534.
- Caprara, G. V., Vecchione, M., Alessandri, G., Gerbino, M., and Barbaranelli, C. (2011). The contribution of personality traits and self-efficacy beliefs to academic achievement: A longitudinal study. *British Journal of Educational Psychology*, 81:78–96.
- Chalmer, P., Pritikin, J., Robitzsch, A., Kim, K., Falk, C. F., and Meade, A. (2017). Package 'mirt'.
- Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, 48(6):1–29.
- Chamorro-Premuzic, T. and Furnham, A. (2014). *Personality and Intellectual Competence*. Psychology Press, New York, New York, USA.
- Champagne, A. B. (1980). Factors influencing the learning of classical mechanics. American Journal of Physics, 48(12).
- Chen, G., Gully, S. M., and Eden, D. O. V. (2004). General self-efficacy and self-esteem: toward theoretical and empirical distinction between correlated self-evaluations. *Journal* of Organizational Behavior, 25:375–395.
- Cohoon, J. (2003). Must there be so few? Including women in CS. Proceedings of the25th International Conference on Software Engineering., pages 668–674.
- Connor, M. C. O. and Paunonen, S. V. (2007). Big Five personality predictors of postsecondary academic performance. *Personality and Individual Differences*, 43:971–990.
- Corno, L. (1986). The metacognitive control components of self-regulated learning. *Contemporary Educational Psychology*, 11(4):333–346.
- Credé, M. and Phillips, L. A. (2011). A meta-analytic review of the Motivated Strategies for Learning Questionnaire. *Learning and Individual Differences*, 21:337–346.
- Danielsiek, H., Toma, L., and Vahrenhold, J. (2017). An Instrument to Assess Self-Efficacy in Introductory Algorithms Courses. Proceedings of the 2017 ACM Conference on International Computing Education Research, pages 217–225.
- Dettmers, S., Trautwein, U., Lüdtke, O., Goetz, T., Frenzel, A. C., and Pekrun, R. (2011). Students' emotions during homework in mathematics: Testing a theoretical model of antecedents and achievement outcomes. *Contemporary Educational Psychology*, 36(1):25– 35.
- Eckerdal, A., Mccartney, R., Moström, J. E., Sanders, K., Thomas, L., and Zander, C. (2007). From Limen to Lumen: Computing students in limital spaces. *Proceedings of the* third international workshop on Computing education research, pages 123–132.
- Evans, G. E. (1988). Learning to program computers by learning to solve problems. *Journal* of Education for Business, 64(2):77–79.
- Falkner, N., Vivian, R., Piper, D., and Falkner, K. (2014). Increasing the effectiveness of automated assessment by increasing marking granularity and feedback units. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*, pages 9–14, New York, New York, USA. ACM Press.
- Fan, X. and Sun, S. (2013). Item Response Theory. In Teo, T., editor, Handbook of Quantitative Methods for Educational Research, pages 45–70. Sense, Rotterdam.
- Ford, J. K., Smith, E. M., Weissbein, D. A., Gully, S. M., and Salas, E. (1998). Relationships of Goal Orientation, Metacognitive Activity, and Practice Strategies With Learning Outcomes and Transfer. *Journal of Applied Psychology*, 83(2):218–233.

- French, D. P., Olander, E. K., Chisholm, A., and Mc Sharry, J. (2014). Which Behaviour Change Techniques Are Most Effective at Increasing Older Adults' Self-Efficacy and Physical Activity Behaviour? A Systematic Review. Annals of Behavioral Medicine, 48(2):225– 234.
- Furnham, A. and Chamorro-Premuzic, T. (2004). Personality and intelligence as predictors of creativity. *Personality and Individual Differences*, 37:943–955.
- Gal, I. and Ginsburg, L. (1994). The Role of Beliefs and Attitudes in Learning Statistics: Towards an Assessment Framework. *Journal of Statistics Education*, 2(2):1–15.
- Garson, G. (2008). Path analysis.
- Gecas, V. (1989). The Social Psychology of Self-Efficacy. Annual Review of Sociology, 15:291–316.
- Gibbons, R. D. and Hedeker, D. R. (1992). Full-Information Item Bi-Factor Analysis. Psychometrika, 57(3):423–436.
- Gist, M. E. (1992). Self-Efficacy: A Theoretical Analysis of its Determinants and Malleability. Academy of Management Review, 17(2).
- Gnambs, T. (2015). What makes a computer wiz? Linking personality traits and programming aptitude. *Journal of Research in Personality*, 58(0):31–34.
- Goetz, T., Nett, U. E., Martiny, S. E., Hall, N. C., Pekrun, R., Dettmers, S., and Trautwein, U. (2012). Students' emotions during homework: Structures, self-concept antecedents, and achievement outcomes. *Learning and Individual Differences*, 22(2):225–234.
- Goldberg, L. R. (1990). An Alternative "Description of Personality": The Big-Five Factor Structure. Journal of Personality and Social Psychology, 59(6):1216–1229.
- Goldberg, L. R. (1992). The Development of Markers for the Big-Five Factor Structure. *Psychological Assessment*, 4(1):26–42.
- Goldberg, L. R. (1999). A broad-bandwidth, public-domain, personality inventory measuring the lower-level facets of several Five-Factor models. In Merveilde, I., Deary, I., De Fruyt, F., and Ostendorf, F., editors, *Personality Psychology in Europe (Vol 7)*, pages 7–28. Tilburg University Press.
- Gore, P. A. (2006). Academic Self-Efficacy as a Predictor of College Outcomes: Two Incremental Validity Studies. *Journal of Career Assessment*, 14(1):92–115.
- Gow, A. J., Whiteman, M. C., Pattie, A., and Deary, I. J. (2005). Goldberg's 'IPIP' Big-Five factor markers: Internal consistency and concurrent validation in Scotland. *Personality and Individual Differences*, 39:317–329.
- Hannula, M. S. (2015). Emotions in Problem Solving. In Selected Regular Lectures from the 12th International Congress on Mathematical Education, pages 269–288. Springer International Publishing.

- Havenga, M. (2015). The role of metacognitive skills in solving object-oriented programming problems: a case study. The Journal for Transdisciplinary Research in Southern Africa, 11(1):133–147.
- Hostetler, T. R. (1983). Predicting Student Success in an Introductory Programming Course. *SIGCSE Bulletin*, pages 40–44.
- Hox, J. and Bechger, T. (2007). Introduction to Structural Equation Modeling. Family Science Review, 11(3):354–373.
- Jenkins, T. (2002). On the Difficulty of Learning to Program. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, pages 53–58.
- Judge, T. A., Erez, A., Bono, J. E., and Thoresen, C. J. (2002). Are Measures of Self-Esteem, Neuroticism, Locus of Control, and Generalized Self-Efficacy Indicators of a Common Core Construct? *Journal of Personality and Social Psychology*, 83(3):693–710.
- Judge, T. A., Higgins, C. A., Thoresen, C. J., and Barrick, M. R. (1999). The Big Five personality traits, general mental ability, and career success across the life span. *Personnel Psychology*, 52(2):621–651.
- Kanfer, R. and Ackerman, P. L. (1989). Motivation and Cognitive Abilities: An Integrative/ Aptitude-Treatment Interaction Approach to Skill Acquisition. *Journal of Applied Psychology*, 74(4):657–690.
- Kang, S.-M. and Waller, N. G. (2005). Moderated Multiple Regression, Spurious Interaction Effects, and IRT. Applied Psychological Measurement, 29(2):87–105.
- Kappe, R. and Van Der Flier, H. (2012). Predicting academic success in higher education: What's more important than being smart? European Journal of Psychology of Education, 27(4):605–619.
- Karwowski, M., Lebuda, I., Wisniewska, E. W. A., and Gralewski, J. (2013). Big Five Personality Traits as the Predictors of Creative Self-Efficacy and Creative Personal Identity: Does Gender Matter? *Journal of Creative Behavior*, 47:215–232.
- Kingston, J. A. and Lyddy, F. (2013). Self-efficacy and short-term memory capacity as predictors of proportional reasoning. *Learning and Individual Differences*, 26:185–190.
- Kinnunen, P. and Simon, B. (2010). Experiencing Programming Assignments in CS1: The Emotional Toll. Proceedings of the Sixth international workshop on Computing education research (ICER '10), pages 77–85.
- Kinnunen, P. and Simon, B. (2011). CS Majors' Self-Efficacy Perceptions in CS1: Results in Light of Social Cognitive Theory. Proceedings of the seventh international workshop on Computing education research - ICER '11, pages 19–26.
- Kline, R. B. (2015). *Principles and practice of structural equation modeling*. Guilford publications., 4th edition.

- Komarraju, M., Karau, S. J., and Schmeck, R. R. (2009). Role of the Big Five personality traits in predicting college students' academic motivation and achievement. *Learning and Individual Differences*, 19(1):47–52.
- Kurtz, B. L. (1980). Investigating the relationship between the development of abstract reasoning and performance in an introductory programming class. *ACM SIGCSE Bulletin*, 12(1):110–117.
- Le Pine, J. A., Colquitt, J. A., and Erez, A. (2000). Adaptability to Changing Task Contexts: Effects of General Cognitive Ability, Conscientiousness, and Openness to Experience. *Personnel Psychology*, 53:563–593.
- Lee, W., Lee, M.-j., and Bong, M. (2014). Testing interest and self-efficacy as predictors of academic self-regulation and achievement. *Contemporary Educational Psychology*, 39(2):86–99.
- Lent, R. W. and Hackett, G. (1994). Sociocognitive mechanisms of personal agency in career development: Pantheoretical prospects. In Savikas, M. and Lent, R. W., editors, *Convergence in career development theories: Implications for science and practice*, pages 77–101. CPP Books, Palo Alto, CA.
- Lewis, S. E. and Lewis, J. E. (2008). Seeking effectiveness and equity in a large college chemistry course: An HLM investigation of peer-led guided inquiry. *Journal of Research in Science Teaching*, 45(7):794–811.
- Li, Y., Bolt, D. M., and Fu, J. (2006). A Comparison of Alternative Models for Testlets. Applied Psychological Measurement, 30(1):3–21.
- Lim, B. C. and Ployhart, R. E. (2006). Assessing the convergent and discriminant validity of Goldberg's international personality item pool A multitrait-multimethod examination. *Organizational Research Methods*, 9(1):29–54.
- Linnenbrink-Garcia, L. and Pekrun, R. (2011). Students' emotions and academic engagement: Introduction to the special issue. *Contemporary Educational Psychology*, 36(1):1–3.
- Lishinski, A., Yadav, A., and Enbody, R. (2017a). Gender Differences in Personality, Problem Solving, and Performance in Introductory Programming.
- Lishinski, A., Yadav, A., and Enbody, R. (2017b). Students' Emotional Reactions to Programming Projects in Introduction to Programming: Measurement Approach and Influence on Learning Outcomes. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 30–38.
- Lishinski, A., Yadav, A., Enbody, R., and Good, J. (2016a). The Influence of Problem Solving Abilities on Students' Performance on Different Assessment Tasks in CS1. Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16).

- Lishinski, A., Yadav, A., Good, J., and Enbody, R. (2016b). Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals and Self-Efficacy on Performance. Proceedings of the 12th Annual International ACM Conference on International Computing Education Research (ICER '16).
- Locke, E. A. and Latham, G. P. (1990). A Theory of Goal Setting and Task Performance. Prentice-Hall, Englewood Cliffs, NJ.
- Locke, E. A. and Latham, G. P. (2012). New Developments in Goal Setting and Task Performance. Routledge.
- Lopez, M., Whalley, J., Robbins, P., and Lister, R. (2008). Relationships Between Reading, Tracing and Writing Skills in Introductory Programming. *Proceedings of the fourth international workshop on Computing education research - ICER '08*, pages 101–112.
- Maddux, J. E., editor (1995). Self-efficacy, adaptation, and adjustment: theory, research, and application. Plenum Press, New York.
- Mancy, R. and Reid, N. (2004). Aspects of Cognitive Style and Programming. 16th Workshop of the Psychology of Programming Interest Group, pages 1–9.
- Margolis, H. and McCabe, P. P. (2006). Improving Self-Efficacy and Motivation: What to Do, What to Say. *Intervention in School and clinic*, 41(4):218–227.
- Mayer, R. E. (1992). *Thinking, problem solving, cognition.* WH Freeman/Times Books/Henry Holt & Co.
- Mayer, R. E., Dyck, J. L., and Vilberg, W. (1986). Learning to program and learning to think: what's the connection? *Communications of the ACM*, 29(7):605–610.
- Mccoy, L. P. and Orey, M. A. (1988). Computer Programming and General Problem Solving by Secondary Students Computer Programming and General Problem Solving by Secondary Students. *Computers in the Schools*, 4:151–158.
- Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., Clair, C. S., and Thomas, L. (2006). A cognitive approach to identifying measurable milestones for programming skill acquisition. *ACM SIGCSE Bulletin*, 38(4):182.
- Mischel, W. (2013). *Personality and Assessment*. Psychology Press.
- Multon, K. D., Brown, S. D., and Lent, R. W. (1991). Relation of self-efficacy beliefs to academic outcomes: A meta-analytic investigation. *Journal of Counseling Psychology*, 38(1):30–38.
- Mumford, M. D., Baughman, W. A., Threlfall, V. K., Uhlman, C. E., and Costanza, D. P. (1993). Personality, Adaptability, and Performance: Performance on Well-Defined Problem Solving Tasks. *Human Performance*, 6(3):241–285.

- Nowaczyk, R. H. (1984). The relationship of problem-solving ability and course performance among novice programmers. *International Journal of Man-Machine Studies*, 21(2):149– 160.
- Osborne, J. W. (2002). The effects of minimum values on data transformations. Annual Meeting of the American Educational Research Association.
- Pajares, F. (1996). Self-Efficacy Beliefs and Mathematical Problem-Solving of Gifted Students. Contemporary educational psychology, 21(4):325–44.
- Pajares, F. and Miller, M. D. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of educational psychology*, 86(2):193.
- Pajares, F. and Valiante, G. (1997). Influence of Self-Efficacy on Elementary Students' Writing. Journal of Educational Research, 90(6):353–360.
- Pajares, F. and Valiante, G. (2012). Influence of Self-Efficacy on Elementary Students' Writing. The Journal of Educational Research, 90(6):353–360.
- Palumbo, D. (1990). Programming Language / Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research*, 60(1):65.
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Uni, J. M., Bennedsen, J., West, I. T. U., Devlin, M., and Paterson, J. (2007). A Survey of Literature on the Teaching of Introductory Programming. ACM SIGCSE Bulletin, 39(4):202–223.
- Pintrich, P. R. and De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, 82(1):33–40.
- Pintrich, P. R. and Schunk, D. H. (1995). Motivation in education: Theory, research, and applications. Prentice-Hall, Englewood Cliffs, NJ.
- Pintrich, P. R., Smith, D. A. F., Garcia, T., and Mckeachie, W. J. (1993). Reliability and Predictive Validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and Psychological Measurement*, 53(3):801–813.
- PISA (2003). The PISA 2003 Assessment Framework. Technical report, Organisation for Economic Co-operation and Development.
- Prat-Sala, M. and Redford, P. (2012). Writing essays: does self-efficacy matter? The relationship between self-efficacy in reading and in writing and undergraduate students' performance in essay writing. *Educational Psychology*, 32(1):9–20.
- R Core Team (2017). R: A Language and Environment for Statistical Computing.
- Ramalingam, V., LaBelle, D., and Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '04, pages 171–175.

- Ramalingam, V. and Wiedenbeck, S. (1999). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research*, 19(4):367–381.
- Rammstedt, B., Goldberg, L. R., and Borg, I. (2010). The measurement equivalence of Big-Five factor markers for persons with different levels of education. *Journal of Research* in Personality, 44(1):53–61.
- Raykov, T. and Marcoulides, G. A. (2006). A First Course in Structural Equation Modeling. Lawrence Erlbaum Associates, London.
- Reise, S. P., Ainsworth, A. T., and Haviland, M. G. (2005). Item response theory: Fundamentals, applications, and promise in psychological research. *Current Directions in Psychological Science*, 14(2):95–101.
- Rivers, K., Harpstead, E., and Koedinger, K. (2016). Learning Curve Analysis for Programming: Which Concepts do Students Struggle With? Proceedings of the 12th International Computing Education Research Conference, pages 143–151.
- Robins, A. (2010). Learning edge momentum: a new account of outcomes in CS1. Computer Science Education, 20(1):37–71.
- Robins, A., Rountree, J., and Rountree, N. (2003). Learning and Teaching Programming : A Review and Discussion. *Computer Science Education*, 13(2):137–172.
- Salleh, N., Mendes, E., Grundy, J., and Burch, G. (2009). An empirical study of the effects of personality in pair programming using the five-factor model. 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009., pages 214–225.
- Schunk, D. H. (1995). Self-Efficacy and Education and Instruction. In Self-Efficacy, Adaptation, and Adjustment: Theory Research and Application, pages 281–303. Springer US.
- Schunk, D. H. and Ertmer, P. A. (2000). Self-regulation and academic learning: Self-efficacy enhancing interventions. In Boekaerts, M., Pintrich, P. R., and Zeidner, M., editors, *Handbook of self-regulation*, pages 631–649.
- Selig, J. P. and Little, T. D. (2012). Autoregressive and cross-lagged panel analysis for longitudinal data. In Laursen, B., Little, T. D., and Card, N. A., editors, *Handbook of Developmental Research Methods*, pages 265–278. Guilford Press.
- Sleeman, D. (1986). The Challenges of Teaching Computer Programming. Communications of the ACM, 29(9):840–841.
- Thoms, P., Moore, K. S., and Scott, K. S. (1996). The Relationship between Self-Efficacy for Participating in Self-Managed Work Groups and the Big Five Personality Dimensions. *Journal of Organizational Behavior*, 17(4):349–362.
- Valentine, J. C., Dubois, D. L., and Cooper, H. (2004). The Relation Between Self-Beliefs and Academic Achievement: A Meta-Analytic Review. *Educational Psychologist*, 39:111–133.

- Vermetten, Y. J., Lodewijks, H. G., and Vermunt, J. D. (2001). The Role of Personality Traits and Goal Orientations in Strategy Use. *Contemporary Educational Psychology*, 26:149–170.
- Vermunt, J. D. (2005). Relations between student learning patterns and personal and contextual factors and academic performance. *Higher Education*, 49(3):205–234.
- Waschle, K., Allgaier, A., Lachner, A., Fink, S., and Nuckles, M. (2014). Procrastination and self-efficacy: Tracing vicious and virtuous circles in self-regulated learning. *Learning* and Instruction, 29:103–114.
- Watson, C. and Li, F. W. B. (2014). Failure Rates in Introductory Programming Revisited. Proceedings of the 2014 conference on Innovation & technology in computer science education, pages 39–44.
- Watson, C., Li, F. W. B., and Godwin, J. L. (2014). No Tests Required: Comparing Traditional and Dynamic Predictors of Programming Success. Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14).
- Wiedenbeck, S. (2005). Factors affecting the success of non-majors in learning to program. Proceedings of the 2005 international workshop on Computing education research - ICER '05, pages 13–24.
- Williams, T. and Williams, K. (2010). Self-efficacy and performance in mathematics: Reciprocal determinism in 33 nations. *Journal of Educational Psychology*, 102(2):453–466.
- Wilson, B. C. (2010). A Study of Factors Promoting Success in Computer Science Including Gender Differences. *Computer Science Education*, 12(1-2):141–164.
- Wilson, B. C. and Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. ACM SIGCSE Bulletin, 33(1):184–188.
- Winne, P. H. (1996). A metacognitive view of individual differences in self-regulated learning. Learning and Individual Differences, 8(4):327–353.
- Wolters, C. and Yu, S. (1996). The Relation Between Goal Orientation and Students' Motivational Beliefs and Self-Regulated Learning. *Learning & Individual Differences*, 8(3):211– 238.
- Zheng, L., Goldberg, L. R., Zheng, Y., Zhao, Y., Tang, Y., and Liu, L. (2008). Reliability and concurrent validation of the IPIP Big-Five factor markers in China: Consistencies in factor structure between Internet-obtained heterosexual and homosexual samples. *Personality* and Individual Differences, 45:649–654.
- Zimmerman, B. (2002). Becoming a Self-Regulated Learner: An Overview. *Theory into Practice*, 41(2).
- Zimmerman, B. J. (1986). Becoming a self-regulated learner: Which are the key subprocesses? *Contemporary Educational Psychology*, 11(4):307–313.

- Zimmerman, B. J. (1990). Self-Regulated Learning and Academic Achievement: An Overview. *Educational psychologist*, 25(1):3–17.
- Zimmerman, B. J. and Risemberg, R. (1997). Becoming a Self-Regulated Writer: A Social Cognitive Perspective. Contemporary Education al Psychology, 22:73–101.
- Zimmerman, B. J. and Schunk, D. H. (1989). Self-regulated learning and academic achievement: Theory, research, and practice. Springer-Verlag, New York.
- Zingaro, D. and Porter, L. (2016). Impact of Student Achievement Goals on CS1 Outcomes. Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE '16), pages 279–284.