5-2021

Algorithms for Algorithms: Teaching Problem-Solving in Computer Science

James Riswick-Estelle

Algorithms for Algorithms: Teaching Problem-Solving in Computer Science

by

James Riswick-Estelle

A thesis

submitted to the College of Psychology and Liberal Arts at

Florida Institute of Technology

in partial fulfillment of the requirements

for the degree of

Master of Science

in

Applied Behavior Analysis and Organizational Behavior Management

Melbourne, Florida

May, 2021

We the undersigned committee hereby approve the attached thesis

Algorithms for Algorithms: Teaching Problem-Solving in Computer Science

by

James Riswick-Estelle, B.A.

Kimberly Sloman, Ph.D., BCBA-D
Associate Professor
School of Behavior Analysis
Major Advisor

Rachael Tilka, Ph.D.
Assistant Professor
School of Behavior Analysis
Committee Member

Thomas Eskridge, Ph.D.
Associate Professor
Computer Engineering and Sciences
Outside Committee Member

Catherine Nicholson, Ph.D., BCBA-D
Assistant Professor
School of Behavior Analysis
Committee Member

Kaitlynn Gokey, Ph.D., BCBA-D
Visiting Assistant Professor
School of Behavior Analysis
Committee Member

Robert Taylor, Ph.D.
Professor and Interim Dean
College of Psychology and Liberal Arts

Abstract

Algorithms for Algorithms: Teaching problem-solving in Computer Science

By

James Riswick-Estelle B.A.

Major Advisor: Kimberly Sloman, Ph.D., BCBA-D

Increased demand from society for computer scientists and software engineers has placed considerable stress on university-based computer science and engineering programs. Given technology's central role in society, the education of those developing and maintaining that technology is critical. Behavior-based teaching methods may assist in addressing increased demand on universities and improve the quality of education they provide. The present study included two experiments of non-concurrent multiple baseline design. The experiments included 29 total participants to evaluate different algorithm-writing teaching methods at the undergraduate level. Algorithms describe a problem's key features and outline the step-by-step process required for solving that particular problem. The instructional techniques evaluated included traditional university courses such as textbooks and lectures. During Experiment 1, researchers compared these methods with behavior analytic methods such as task analyses and behavioral skills training. Experiment 2 was a direct replication of Experiment 1 with a more comprehensive task analysis. In both experiments, most participants only displayed significant improvement in the Task Analysis + Behavioral Skills Training phase. In that phase, participant performance improved drastically, generalized to more complex tasks, and maintained several weeks after training.

Table of Contents

List of Keywords

Algorithm

Behavior analysis

Behavioral skills training

College instruction

Computer science

Problem solving

Task analysis

Task clarification

Undergraduate

List of Figures

List of Tables

Acknowledgment

I wish to express my most profound appreciation to those I would call family.

Thank you to my partners, family, friends, fraternity brothers, colleagues, and classmates for your support. Thank you to my committee for your patience and guidance. Special thanks to Dr. Ada Celeste Harvey, Dr. Heather Crawford, and Dr. Kimberly Sloman, whose mentorship and insight I could not have gone without during this project. Most importantly, thank you to Dr. Kaitlynn Gokey. Your mentorship means the world to me, and without it, I do not know how I would have completed this project. You have taught me so much and shaped who I am as a scientist.

Thank you.

Algorithms for Algorithms: Teaching Problem-Solving in Computer Science

Chapter 1: Introduction

The Demand for Education

In the United States, enrollment in post-secondary education has increased by 6% since 1993. By 2018, 2.2 million (69.1%) of the 3.2 million high school graduates between the ages of 16-24 enrolled in post-secondary education (Bureau of Labor, 2017). Commensurate with the rise in college enrollment, university-based computer science and engineering departments have seen a 368% increase in computer science majors from 2006-2017. Additionally, the Bureau of Labor and Statistics expects the market to grow an additional 12% from 2018-2028, adding 546,200 new computer science-based jobs (Bureau of Labor Statistics, 2019; Shein, 2019). Based on these statistics, universities could lose tens to hundreds of millions of dollars in tuition if they cannot attract and maintain new students (Kaplan, 2020).

Science, technology, engineering, and mathematics (STEM) fields carry significant weight and momentum. This momentum is due to how quickly STEM fields make discoveries and how discoveries impact society and industry. This impact is especially salient for computer science because of how integral technology has become in other areas of STEM and society. Undergraduates, therefore, seek out programs that offer high-quality education and up-to-date information from within the industry. Learning a preferred programming language and gaining experience developing projects important to the larger computer science community make undergraduate students highly marketable assets to future employers. In turn, prospective employers seek candidates with these skills, training, and experiences as they need new hires to have competency in these areas. This increased market demand requires universities to consider the number of students they accept into their programs as well as the quality of education and experiences they will provide for enrolled students.

Many events may drive students from a university. Suppose students are not sufficiently supported or find that the quality of education is not adequate. In that case, they may choose to transfer or withdraw from the university entirely. Even when the instruction is of high quality, its benefit is lost when equipment is out of date. The information that students learn is no longer relevant. The education quality could potentially cost universities hundreds of millions of dollars in losses due to lack of student retention after admission (Watson, 2016). Student attrition incurs costs similar to those of low application or admission to a program. Therefore, student attrition indicates that it is not enough to get students to apply and enroll but that universities must maintain and support those students as a priority.

Solutions

Universities most commonly employ at least one of three approaches to meet market demand and mitigate market pressures. The first approach is to expand class sizes. The second is to hire additional faculty and increase the number of sections within a course. Lastly, as an alternative to hiring additional faculty, universities may employ graduate students to teach courses as part of their degree program (Shein, 2019). While one or several of these methods may appear to reduce departmental strain, the reality is that the strain is merely shifted rather than adequately addressed.

Increase Class Sizes

Expanding class sizes is often the first approach used by universities. It requires the least amount of additional resources from the university, namely providing the physical class space and faculty resources to manage and grade additional students. However, there is no guarantee that universities will provide additional time or resources to faculty to adjust for additional students and increased professor workload. If there are no additional resources, there is then an increased risk for faculty burnout and turnover. Faculty burnout or turnover, in turn, could compound circumstances as the workload of any faculty lost is redistributed amongst the remaining faculty.

Increased class sizes have an additional downside, as students tend to prefer smaller class sizes (Feld & Grofman, 1977). Smaller class sizes often allow professors to easily include exercises, activities, and discussions requiring students to engage with the current course material. Larger class sizes (i.e., 50+) are typically lecture-exclusive, which do not require much involvement from the student or much immediate use of the current course material. The lack of engagement opportunities understandably makes the classes less appealing to students and leads to lower attendance rates (Feld & Grofman, 1977).

The impact of student participation on student performance is consistent with Kim et al. (2019). Kim et al. explored the correlation between attendance and higher academic performance, finding that attendance did not significantly impact student performance when controlling for participation. These findings indicate that class participation significantly influences student performance and directly impacts the value of attending a class (Kim et al., 2019). Therefore, an increase in class size could lead to a decline in faculty and student performance.

Recruit Additional Faculty

Rather than increasing the number of students in a class, universities may recruit additional faculty to offer more class sections. This strategy allows for smaller class sizes and smaller student-to-teacher ratios. Smaller student-to-teacher ratios should increase student satisfaction and engagement as well as the quality of instruction (Feld & Grofman, 1977). However, this means an increase in the demand for various university resources. University resources may include faculty salaries, campus spaces, classrooms, computers, as well as support staff, facilities, and services. An increase in the demand for any of the university resources mentioned above could increase tuition for students as universities find ways to mitigate expenses.

Regardless, while students may prefer and generally perform better in smaller class sizes, there can be significant variability in professor quality and performance (Feld & Grofman, 1977). The variability of professor quality could be because most professors have no formal training in teaching and instructional technology (Skinner, 1968). That is to say, smaller class sizes may mitigate deficits in instructional skills and allow for acceptable outcomes, but they do not directly address skill deficits. Considering that professors are highly educated and highly specialized, universities expect professors to know how to instruct merely by being asked to do so. In other words, universities assume professors already have these skills. Alternatively, if they do not, they will develop them organically and be self-taught in the absence of formal training (Fertig, 2012).

Even when universities decide to hire additional faculty, it can be challenging to hire faculty at a rate that matches growth, primarily when growth occurs at high rates. Challenges with hiring additional faculty are compounded within computer science as universities struggle to compete with other industry professions. Universities struggle to compete because other industries often pay at least twice to three times as much as university faculty positions. Additionally, other positions typically offer more significant benefits and bonuses when compared to universities (Shein, 2019). Lastly, weak competition with other industries creates situations where the computer scientists hired by universities are potentially less competent than computer scientists hired elsewhere in the industry.

Graduate Student Instructors

Graduate student instructors offer an alternative to hiring additional faculty. Graduate student instructors reduce strain on current professors while still providing additional support for students. They also allow university programs to increase enrollment without incurring high additional costs. Using graduate student instructors also creates opportunities for those graduate students to earn a wage while in graduate school and gain teaching experience at the university level. However, this solution creates potential conflicts of interest for graduate students as they take on a dual role of being both a student and an employee of the university.

The drawbacks to being in this dual role can include additional work that may compete with work needed for their degree. Changes in their relationships with other students and faculty, primarily as they conduct classes and office hours, may create a conflict of interest. Graduate students will have to choose between time and resources spent towards benefitting one role over another. Additionally, this solution does not address the quality of instruction students may receive. Students may receive or perceive they are receiving instruction from someone who is not trained in teaching and not fully established in their field (Shein, 2019).

Impact Considerations

The options of increased class sizes, recruitment of additional faculty, and the implementation of graduate student instructors do not, even in combination, offer a perfect solution to the challenges faced by universities. Therefore, it is crucial to consider metrics such as timeliness, cost, quantity, and quality. These metrics are crucial because they allow for a better understanding of how these decisions will impact the people within and around the organizations. Additionally, these metrics are essential because they provide more precision when evaluating these decisions' long-term practicality (Daniels & Bailey, 2004).

Timeliness

Timeliness refers primarily to when a critical activity will be complete. Universities have consistent hard deadlines regarding enrollment that have long-term effects if they cannot meet demand and provide enough placement for incoming students. Delays could mean losing many potential new students to other universities. For example, suppose a university decides to accept a student but does not extend an offer in a timely manner. In that case, another university could extend an offer during this time and have it accepted by the student. Delays could also have effects beyond a student's initial enrollment at a university, causing recurring disruptions resulting in attrition due to poor ongoing scheduling and planning. For example, suppose two universities accept the same student. In that case, the student may select the university that provides better accommodations towards their schedule and graduation timeline.

Cost

Cost refers to the investment of money, labor, materials, management, and other resources required to produce goods or services (Daniels & Bailey, 2004). The cost of expanding class sizes or hiring a graduate student to instruct is much more cost-effective than hiring additional full-time faculty. For example, full-time faculty salaries ($60,000-$100,000) are at least two to 4 times more than that of a graduate student assistant ($15,000-$30,000; Academic Positions, 2018). However, this may cost universities as students may respond negatively to crowded classrooms, underequipped courses, or the perception that less qualified instructors teach them.

Quantity

For universities, quantity often refers to the number of students who applied compared to how many were accepted or how many students enroll in a course compared to the resulting class grade distribution. It may also refer to how many students graduate to become successfully employed in their major. The element of quantity can produce undesirable consequences if not measured carefully. For example, taking the number of students who graduate without considering how long it took them to graduate or how long it takes for graduates to find employment in their major. Students may graduate on time but be unable to find a job for a considerable length of time. These consequences can occur if, for instance, the number of students taking courses and graduating looks favorable. However, the system's favorability is lost when moving a large number of students through that system is not leading to desired long-term results.

Quality

Quality refers to the extent to which a product or service meets or exceeds an expectation or standard. The three elements of quality include accuracy, rank, and novelty. First, accuracy is defined by how well the product fulfills the expectation or standard. Second, rank or class refers to how much the product exceeds the expectation or standard. Lastly, novelty describes how the product builds onto or adds to the expectation or standard (Daniels & Bailey, 2004). In the context of university education, the goal of any student is to gain the education required to start a fulfilling career within a particular field or specialty.

Accuracy. Universities need to have the capability to deliver a basic standard of education that any student could reasonably expect. Accreditation bodies primarily control and set most universities' standards so that educational institutions are guaranteed to provide a certain level of quality (U.S. Department of Education, 2020). Accreditation can vary but generally ensures that specific topics are present and conveyed in a particular way. Accreditation acts as a protective measure to assure students, financial institutions, and other vested parties. It indicates that universities act in good faith with the law, provide a certain minimum quality of education, and conduct business as advertised. However, accreditation cannot account for the many variations within and between universities that result in the vast differences in rankings, reputations, research, and outcomes for students.

Rank. A university's rank can vary depending on what tools and metrics an assessment implements. Within educational quality, things like student performance, graduation rates, alumni employment, student to faculty ratios, and student opportunities influence rank (Studyportal, 2021). Rank is where universities build beyond a minimum standard and where students most often look when searching for a university. Rank is an essential consideration to students. Rank is essential because it will most likely have the most considerable impact on a student's primary goal: to graduate on time and start a successful career.

Novelty. Universities are novel based on what they do to make themselves unique and stand out amongst other universities. Novelty amongst universities can include many different things: location, recreational opportunities, and proximity to potential employers. Additionally, what makes a university novel could be related to unique facilities and equipment or renowned faculty and research teams. Several universities may deliver high-quality education, but not all can distinguish themselves as the only university or program to offer particular courses. Alternatively, a university may be the only institution to have unique partnerships with hiring businesses, which allow students easy access to internships and work opportunities.

Beyond University

After a student graduates from university, the quality of the education they received becomes much more critical. For example, the importance of quality assurance with computer systems becomes evident when errors incur a significant loss of life and financial cost. Consider the October 2018 and March 2019 crashes of two Boeing 737 Max jets. These crashes were due to a bug in an automated system designed to regulate engine performance and avoid stalls regardless of pilot input (Gelles, 2019). These two crashes lead to a total of 346 deaths as well as upwards of over seven billion dollars worth of losses for Boeing™ (Gelles, 2019).

These avoidable plane accidents show that the program implemented solved one problem (i.e., regulating flight performance) without considering other potential problems in the broader situation. The lack of consideration for how this program could affect other variables created additional and separate problems from the first. A well-developed algorithmic process would have considered the full scope of the situation and the impact all parts may have. The value these components have is why they are highly valuable in other areas such as system engineering and organizational behavior management. These two accidents are just one instance of hundreds of others that have cost thousands of lives and billions of dollars over a few decades (Kienitz, 2019).

Behavior Analysis, Algorithms, & Problem Solving

The subject of computer science is broadly and primarily about problem-solving, algorithmic thinking, and data structures. While the critical content is well established within computer science, how to teach it varies considerably (Baeza-Yates, 1995). What this typically translates to in the classroom is an approach to teaching based on exposing and instructing students to solve various computer science problems. These problems can include many different features, functions, classes, and types that provide different examples of various applications of techniques and strategies (Hanly & Koffman, 2016).

Algorithms

Algorithms are a formulaic approach to solving a given problem. A complete algorithm includes several instructions or methods implemented to solve a problem. Algorithms are clear and precise, define inputs and outputs, and have established and measurable parameters (Crawford, 2019; Hanly et al., 2016). An example of the kinds of problems a student might be required to solve using an algorithmic approach could include database management. Database management would involve being provided a database of information and having to sort and organize it. The task may also include identifying, selecting, and manipulating certain portions of information from that database.

A database could be an Excel® spreadsheet, Word® document, or many different files. A programmer would first need to decide which programming language would be best to accomplish the task. Then the scope of the problem needs to be considered and defined. The scope can include what kind of information gets read in and then printed out by the program (i.e., whole numbers, fractions, letters, words, sentences, etc.). The scope may also include what operations or manipulations need to be performed on the input to reach the desired output, as well as how and what tools or functional methods the programmer will use to accomplish those operations. Lastly, the programmer needs to test and evaluate their program for functionality and efficiency.

These steps present several different problems with many other sub-problems, which all need to be solved. Solving requires analysis, planning, note-taking, and experimentation. The ultimate result of this process of steps is an algorithm that solves the problem. It is possible to develop an algorithm for any problem, even a sub-problem within a more significant problem. Programming something like a calculator can sound straightforward. However, developing a calculator requires keeping track of the numerical and symbolic input in a particular order, performing the correct operation, and outputting an accurate result. These are just a few details the programmer must consider. However, designing a calculator can involve many more. Accounting for these considerations requires a lot of skill, problem-solving, and planning to complete.

The issue with requiring students to primarily solve problems as the primary approach to teaching algorithm development is that it is a brute force approach. The problem with a brute force approach is that it does not precisely target or sequentially build up a students' problem-solving skills. It may require students to engage in relevant behaviors but does not do so systematically. It also does not necessarily require students to learn best practices. Lastly, it can fail to teach students to identify what elements are specifically crucial to evaluating and fully understanding a problem. This approach's deficits leave students to navigate potential gaps in their background knowledge and experience or make leaps in logic by themselves. While a brute force approach can be successful for some, there are many others for whom it is not. Less fortunate students may be left developing more costly and less efficient algorithms than they otherwise could be.

Behavior Analysis

Behavior analysis provides additional supports and research-based teaching methods that can ensure a high quality of instruction for computer science courses (Sulzer-Azaroff, 1986). Behavior analysis, a natural science of learning and behavior, offers practical and empirically evaluated instructional methods that could significantly impact computer science. Behavior analysis focuses on the experimental arrangement of variables in the environment, observable and measurable qualities of behavior, and the means of accurately predicting behavior under specific environmental arrangements. The scientific approach of behavior analysis has led to significant contributions to education, medicine, and business (Baer et al., 1968).

Behavior analysis has a long history of successful teaching, coaching, and training going as far back as the 1950s (Sulzer-Azaroff, 1986). Some of the most well-known methods include precision teaching, programmed instruction, and direct instruction (Gettinger, 1993; Lindsley, 1992; Tudor & Bostow, 1991). Behavior analysis has effectively taught a wide variety of skills across different ages and populations, including those with disabilities (Sulzer-Azaroff, 1986). Not only are behavior analytic approaches effective, but they are also significantly more effective than other teaching methods.

Project Follow Through, "the largest and most expensive social experiment ever launched" (McDaniels, 1975), evaluated 22 educational methods with thousands of children across hundreds of school districts. Evaluations included several different metrics, including language, spelling, math, reading, self-esteem, and problem-solving skills. Behavior analytic models, like direct instruction, scored 20 to 40 percent higher than the traditional school-based average (Watkins, 1997). By comparison, conventional methods showed no or detrimental effects. These results indicate that behavior analytic methods are high quality, effective, and promptly deliverable to many students at low costs.

Behavior analysis has also been effective at the university level, most notably methods such as the personalized system of instruction, active student responding, and intertech (Keller, 1968; Saville et al., 2006; Sulzer-Azaroff, 1986; Zayac et al., 2016). The critical aspects of behavior analytic methods are not the packaged methods themselves but the core components. Important core components include clear and skill-specific learning objectives with criteria for mastery, well-developed instructions, learner-paced instruction, and the use of positive reinforcement. Other essential features include opportunities for the learner to observe, discuss, and ask questions of a subject matter expert. Lastly, the learner must have the opportunity to frequently engage in the skill, the opportunity for frequent feedback, and the benefit of data inform instruction. If flexibly implemented into a college classroom, these core components could significantly change outcomes for students.

Problem Solving

In any circumstance, it is critical to consider the setting as well as the goals and outcomes when evaluating how a behavior analytic approach can be applied. In a university environment, this means addressing the demands and challenges that create barriers to providing an education. More specifically, this means addressing each student's needs and supporting staff and faculty members. To accomplish this goal, staff and faculty require resources to create an optimal educational environment. These resources allow instruction to be delivered in a timely, cost-effective, and high-quality manner to meet each student's needs. Within computer science, this means providing the tools to teach effective algorithm writing. Algorithm development, ultimately being a thorough and systematic problem-solving process (Denning, 1989; Hanly & Koffman, 2016).

Mayfield & Chase (2002) define problem-solving as a novel combination of several previously learned responses that serves as a response to novel stimuli or set of stimuli (Mayfield & Chase, 2002). Therefore, the first step to effective problem-solving is for the learner to master a set of prerequisite responses required to make novel combinations of those responses in the presence of novel stimuli (Mayfield & Chase, 2002). For example, when performing addition, an individual must be able to respond correctly to the value of different numbers, the difference between values, and the operations involved. The second step is that the learner must have the ability to identify the parameters that define the required response's scope (Mayfield & Chase, 2002; Robbins, 2011). Consider the previous example. If given the option for addition or subtraction, the learner must be able to identify which based on if there is a symbol for addition or subtraction.

Robbins (2011) identified three common mistakes made when teaching problem-solving. The first mistake is that students often receive open-ended problems to solve. Open-ended problems make it difficult for the student to know if they have performed the task correctly and may inadvertently teach incorrect approaches. The second mistake is when the teacher demonstrates problem-solving skills from an expert's standpoint without demonstrating the solving. The solving part is crucial. It includes when the teacher had to experiment and struggle with different combinations to reach the terminal goal. The third and last mistake discussed involves watching peers solve problems. Watching peers solve problems is similar to watching a teacher solve problems. It is not necessarily what is most effective for the learner and does not demonstrate the problem-solving skill (Robbins, 2011).

Learning Objectives

Learning objectives are critical when teaching any skill. For both teachers and students, they provide clear definable goals based on prerequisite skills. They are also fundamental when teaching problem solving because they address the two required components for problem-solving; prerequisites and performance parameters. They also address one of the common problems when teaching problem-solving; open-ended problems. Learning objectives accomplish this because they require assessing what an individual is already capable of doing. Based on what the learner can already do, learning objectives outline what the student needs to learn to accomplish the end task. Learning objectives provide a step-by-step outline of a task with additional information that reduces errors and clarifies the task. Learning objectives also establish learning parameters and often provide examples and non-examples of the target skill to clarify those parameters. While not all students may necessarily meet these targets, learning objectives outline what should be practical and possible for each student. Instructors should develop learning objectives as the first step of course design, assuming prerequisite skills (Beckeschi & Doty, 2000).

Due to their broad applicability, learning objectives belong to a broader class of techniques referred to as task clarification (Crowell et al., 1988). Other applications of task clarification include task analysis, checklists, and job aids. Algorithms can also be considered a kind of task clarification. Algorithms are a form of task clarification because they include many of the same elements found in task clarification. However, algorithms provide more detail as they outline and describe each step in a process, including the starting and ending criteria, and often reference related and contextual material. Regardless of the name, each leads to considerable performance improvements across several different settings (Bacon et al., 1983; Resnick et al., 1973).

However, despite the broad usage of task clarification, Anderson et al. (1988) is one of the only studies to examine the effects of task clarification entirely as an isolated intervention. The researchers used checklists with student employees to improve cleaning performance at a university bar. The checklists improved student cleaning performance by 13%. In a later phase, researchers added a component in which publically posted graphic performance feedback was made available for employees. The posted feedback resulted in an additional 37% increase in performance. The results are consistent with other research studies that combine task clarification and other techniques (Bacon et al., 1983).

The basis for combining task clarification with other techniques is unclear. There is almost no literature in which task clarification is the sole independent variable without some form of additional components, even as just one condition. The lack of research is a stark comparison to the considerable body of literature that includes additional components with task clarification. The lack of isolated literature in this area might be because task clarification is an antecedent-based method. Therefore, as an antecedent-based strategy without a corresponding consequence, it is not expected to lead to enough performance improvements to reach the required performance levels.

Rantz et al. (2009) investigated situations in which checklists alone may not be sufficient. In aviation, a common feature during different flight portions (e.g., before, during, after, etc.) is extensive checklists. Pilots use flight checklists to ensure the proper completion of all flight steps to avoid disastrous and potentially fatal consequences. Despite these checklists' extensive and strict nature in aviation, performance is, broadly, not at acceptable levels. In previous studies, researchers found that around 73% of flight crews committed errors. The errors ranged from 0 to 14 errors per trial with a mean of two (Helmreich et al., 1999, 2001). Therefore, researchers implemented feedback to improve performance with 8 undergraduate flight students. Participants received both technical feedback as well as graphical feedback along with praise for correct performance. Performance improved across all participants to near-perfect scores and maintained during the withdrawal phase (Rantz et al., 2009).

Behavioral Skills Training

Considerable research has evaluated the effects of behavioral skills training across several different skills with various populations. Most notably, the majority of this research involves teaching safety skills to neurodivergent or non-typically developed children. Given the importance of some of these skills on children's health and well-being, it is understandable why this is the case. Despite the amount of research in this area, there is still some discussion about behavioral skills training efficacy. Miltenberger (2008) briefly reviewed several of the studies that added an in-situ or live component of training due to the inability of some skills to generalize to the natural environment through roleplay alone. The necessary implementation of an in-situ element highlights the importance of the training matching the natural environment as closely as possible (Miltenberger, 2008).

Behavioral skills training typically contains four components: instruction, modeling, rehearsal, and feedback (Belisle et al., 2016; Parsons et al., 2012). As a package intervention, behavioral skills training includes several sub-components, including elements of task clarification. Lewon et al. (2019) explore these components in depth while using behavioral skills training to teach researcher skills for researching scent detection with rats. Participants included 4 trainee researchers and eleven rats. Throughout the training, primary-researchers sequentially introduced each component of behavioral skills training to the trainee-researchers. Results indicated that with each component addition, performance increased proportionally until reaching the highest performance with the final component's addition (Lewon et al., 2019). This study shows the value and importance of fully understanding each component and implementing them well.

Amongst adult neurotypical populations, it is not clear if the inclusion of an in-situ component is necessary. An explanation for this could be that not all skills differ significantly outside of a natural environment to warrant requiring in-situ training. Several studies with neurotypical adults demonstrating the efficacy of teaching relatively complex skills using behavioral skills training through roleplay only. Sump et al. (2018) evaluated the effects of training various behavior therapy skills using behavioral skills training to seven previously untrained undergraduate students. The training took place through telehealth and in-person mediums. It included preference assessments, instructional environment management, consequence delivery, and discrete trial training. While studies such a these exist, much more additional research is needed.

Instruction

Instruction is similar to task clarification. Instruction encompasses the skill's full scope with enough information to complete the entire task and respond to most scenarios without significant additional information. The instruction may also include various examples and reasoning as to why the skill is essential. Alternatively, task clarification highlights specific and vital components that may lead to errors if not completed correctly. Instructions are typically limited to vocal or textual verbal stimuli. However, no formal parameters exist that define how to convey the information or how much information is involved. For example, Speelman et al. (2015) presented 4 recreational blackjack players with the following;

> You are about to play blackjack. You will start with $200 worth of chips. Before each hand you will place a bet by placing chips in the circle area in front of you. You may bet as little or as much as you want for each hand; however, you must bet at least the $5 minimum. The goal in blackjack is to have a hand that is closer

> to 21 than the dealer. An ace is valued at 1 or 11. Face cards and tens are valued at 10. All number cards are valued at their face value. The dealer will stand on 17 or higher and hit on 16 or lower. The dealer must hit 17 if they have an ace, known as a "soft" 17. Place bets as if you were playing with real money. (Speelman et al., 2015)

These instructions, the description of the blackjack rules, were read aloud via a video recording. Additionally, the video included instructions, a rationale, and an explanation of the game and its rules (Speelman et al., 2015). This example includes several components and could be reduced to a few sentences or expanded to several pages. Examples such as these show the flexibility and versatility in instruction delivery. This versatility calls for further examination into what counts as instruction or if, at some point, it should be considered something else.

Modeling

Modeling is a demonstration or performance given by a subject-matter expert on the target skill. A subject matter expert has a robust knowledge base of the task and history of fluently performing it. Fluency is the ability to repeatedly perform a skill quickly and effortlessly with high correct performance rates. (Chiesa & Robertson, 2000; Kolmar, 2020). Familiarity and fluency with the task are essential because the subject matter expert must demonstrate the learner's task as it would occur in an optimal live scenario. The modeling should also include various examples close to real-life circumstances and performed as clearly as possible so that no superfluous variables contaminate or distract from the demonstration (Durgin et al., 2014).

Seiverling et al. (2012) used behavioral skills training to train parents to implement a home-based food sensitivity program. The study included 6 participants, 3 mothers with their 3 children diagnosed with autism spectrum disorder. During baseline, parents were presented with a task analysis and asked to conduct the procedure. Later during parent training, researchers reviewed the task analysis with parents then conducted two live modeled sessions of the procedure with the child before conducting the rehearsal. Performance improved immediately and continued to improve with rehearsal and feedback. Afterward, all 3 parents rated the procedure as excellent during a social validity survey, citing modeling as the most helpful (Seiverling et al., 2012). This study demonstrates the proper implementation of modeling and the significant impact high-quality modeling can have on performance and learner satisfaction.

Rehearsal

During rehearsal or practice, learners have an opportunity to practice the skill. Rehearsals repeat until the trainer determines that that learner has achieved and sufficiently demonstrated mastery. Performance should reach high and stable fluency rates before the skill should be considered mastered (Chiesa & Robertson, 2000; Miltenberger, 2008). Nigro-Bruzzi et al. (2010) implemented an extensive roleplay rehearsal phase and live rehearsal phase. The goal was to train 3 special education teachers and 3 speech therapists to implement mand training with 6 children diagnosed with autism spectrum disorder. Manding is the behavior of communicating a desire or a request for something. The first round of training included a baseline with behavioral skills training implementing rehearsal through roleplay. The second round of training included another baseline with live rehearsal.

Nigro-Bruzzi et al. (2010) demonstrate how roleplay-based rehearsal allows the learner to practice the skill before performing that skill in a live situation. Rehearsing through roleplay is helpful because there may be drawbacks to rehearsing in a live situation that may be unpreferred or harmful. For instance, if a trainer makes a mistake during live training, it may delay the correct acquisition of the mand due to time spent correcting the training error. In contrast, the errors may not occur during live training if addressed during roleplay instead. Furthermore, Nigro-Bruzzi et al. (2010) showed roleplay-based training to be highly efficient. Less than 3 sixty-minute roleplay sessions were needed to reach high-performance levels. When participants started the second round of training (live), participants scored higher than they otherwise would have. Live rehearsal then allowed for participants to perform at high levels and generalize in a live scenario in a short amount of time. Ultimately, this study showed the importance of high-quality rehearsal. It also showed how to systematically develop practical and effective staff training to reach more immediate and effective outcomes for clients (Nigro-Bruzzi et al., 2010).

Feedback

After rehearsal, a subject-matter expert provides feedback on what the learner performed correctly and incorrectly, as well as how to improve. Feedback is one of the most common independent variables, especially when combined with task clarification (Crowell et al., 1988). The characteristics of feedback are unclear as several different definitions for the technique exist. What is clear about feedback is that it acts as a stimulus that transmits information to the performer about their performance (Alvero et al., 2001; Crowell et al., 1988; Palmer et al., 2015). The absence of research in this specific area leads to a lack of consensus on feedback's functional properties. The discrepancy regarding whether the feedback is an antecedent or consequence is mainly due to differences in how it can be delivered.

Despite the lack of clarity on the function of feedback, feedback by itself is a very effective teaching strategy (Alvero et al., 2001). Alvero et al. (2001) reviewed feedback during sixty-eight applications across forty-three studies, listing feedback alone as the most commonly used procedure. However, despite this high favorability, it is more effective when combined with other techniques. It is often most effective when combined with antecedent strategies like task clarification (Alvero et al., 2001). This review highlights the importance of addressing performance improvement using multiple combined techniques and the value that feedback contributes and gains as a component of behavioral skills training.

Purpose

Based on the broad application of behavior analysis and its success, as well as the request for a more qualitative competency-based approach in the education system, the adoption of behavior analytic techniques should be well known and widespread (Twyman, 2014). Unfortunately, despite the results of studies like Project Follow Through, behavior analytic approaches have not been widely adopted. Behavior analysis has not effectively disseminated because its philosophies are not well understood. Its methods often appear as though they require lots of time, effort, and training to utilize. (Austin & Soeda, 2009; Sulzer-Azaroff, 1986). These difficulties often make behavior analysis look to run counter to everyday experience, making those who are unfamiliar uncomfortable.

Therefore, the purpose of this study was to evaluate both traditional and behavior analytic teaching methods to improve dissemination and further the adoption of behavior analytic methods in mainstream education. A traditional course instruction style, including textbook-based definitions and lectures, was evaluated and compared with task clarification and behavioral skills training. The use of task clarification and behavioral skills training is due to how they can be developed and packaged for the classroom without significant redesign and disruption of existing courses. Furthermore, computer science presents a particular need given the significant role technology plays in society and the significant challenges university-based computer science departments face.

Chapter 2: Method - Experiment 1

Participants

Participants included 21 undergraduate university students. Recruitment of participants occurred from within the undergraduate CSE1001 Fundamentals of Software Development 1 course. Among the 21 undergraduate participants, 17 were men and 4 were women. All but one participant were between the ages of 16 and 22-years-old. Additionally, all but one participant reported at least one month of programming experience. Most participants had three or more months of programming experience. Researchers placed participants into one of three groups based on selection criteria for a control group and based on current course progression at the time of recruitment during the semester. Group 1 consisted of 14 participants, Group 2 consisted of six participants, and Group 3 consisted of one participant. See Tables 1 and 2 for a summary of participants, demographics, and background information for Groups 1 and 2.

Recruitment

During recruitment, the primary investigator and course instructor presented a PowerPoint slideshow that outlined the study's details. The PowerPoint presentation included a brief overview of behavior analysis, the importance of algorithm writing, the study's purpose, and its relevance to the course. Additionally, the presentation provided details on procedures, the time commitment, potential risks and benefits, and extra credit options. There were no exclusionary criteria related to gender, sex, race, ethnicity, or national origin. The only exclusionary criterion was that students who had previously taken the course were ineligible to participate due to previous exposure to several independent variables.

Extra Credit

The instructor offered 5% extra credit for participating in the study, with an alternative prorated assignment worth an equivalent amount of extra credit available for non-participants as well as for participants who withdrew. Participating in both options for double extra credit (10%) was not available to students. Additionally, the study included three stages corresponding with different independent variables: Baseline, Technical Definition, and the In-Class Lecture comprised stage one. Task Analysis and Task Analysis + Behavioral Skills Training comprised stage two, while two generalization and maintenance probes comprised stage 3. The first and second stages were worth 2% of extra credit, and the third was worth 1%, with the extra credit being contingent on the completion of each stage.

Data Collection

Confidentiality was a significant consideration. Therefore, documentation containing personally identifying information was limited to the primary investigators, co-investigators, and graduate assistants as approved by the primary and co-investigators. Documentation containing personally-identifying information included the following; informed consent forms, emails, appointments, surveys, and data covered by the Health Insurance Portability and Accountability Act or the Family Education Rights and Privacy Act. Access by graduate research assistants was limited to what was required to complete specifically assigned tasks.

Outside of the information in the documents mentioned above, no participant names were present in this study or any obtained data publications. Participant initials were present during data collection then converted to a pseudonym for analysis and presentation purposes. Researchers stored surveys digitally using Google Drive™ and all other documents on BOX™. Researchers destroyed hard copies of documents once digitally stored. Online documents were password protected using two-factor authentication, and access was limited to researchers working on the study.

Setting & Materials

Sessions occurred on a university campus in study areas such as libraries, classrooms, labs, office spaces, study halls, and similar workspaces. At a minimum, these spaces included research materials, a table, and two chairs for the researcher and the participant. The specific time and location of research sessions were selected based on participant and researcher availability and preference. Research sessions were indistinguishable from typical group study activities regarding time, effort, and appearance. Session design was intended to preserve both the participant's privacy and minimize the effort required for participation. Sessions were 30 mins in length, and the researcher would regularly provide an opportunity for 10 min breaks. Breaks were especially significant because most participants requested scheduling back-to-back or extended sessions, resulting in an average cumulative session length of an hour to an hour and a half.

Session materials included a work problem packet, additional information documents, a data collection sheet, and a pen. The work problem packet included a pool of 25 unique problems that researchers would choose a single problem from at the beginning of each trial. The primary researcher and the course instructor collaborated to develop the work problems to match the computer science problems students would have to solve and write algorithms for during the regular course. During each trial, participants wrote their solution on a datasheet provided by researchers.

The data sheet allowed the researcher to collect participant solutions for performance analysis and interobserver agreement. There is a significant amount of planning surrounding computer science problems before writing in code. A well-approached planning phase should result in an algorithm that, when implemented, solves the problem or at least one that comes close. The process is similar to showing work on a math problem; however, algorithms are more complex and abstract. The materials specifically targeted the planning process by requiring participants to use pen and paper. Requiring this approach removed the opportunity to solve the problem by writing code and using a trial-and-error approach. Additionally, the use of a data collection packet allowed discrete collection of data on each trial. Lastly, the additional information documents corresponded to the active independent variable (i.e., technical definition and task analysis). See appendix A for task analysis.

Dependent Variable

The dependent variable was the percentage of steps completed correctly for creating an algorithm. The task analysis defined each of the correct steps. During each trial, researchers would review the task analysis and score how many steps participants completed correctly. The number of correctly completed steps was counted and divided by the total number of steps available. Researchers then multiplied the resulting number by 100 to get a percentage. The mastery criterion was set as 80% correct or higher. The rationale for the mastery criterion was that the study's context focuses on instructional design for a college freshmen-level course. Therefore, the mastery criterion corresponds with the traditional grading system. A traditional grading scale considers 80% or greater (‘B’ and ‘A’) to be within good standing. In contrast, while 60-79% (‘D’ and ‘C’) pass at an undergraduate level, it does not typically indicate fluency with material (NCES, 2011).

The task analysis focused on participants engaging in behaviors critical to assessing and solving a problem. Therefore, even if the solution they developed was not necessarily the best for that particular problem, the participants still completed all the steps necessary for a correct one. Additionally, many of the behaviors that go into problem-solving occur privately. Students either did not always tact or actively identify all the steps they take or skip steps and miss things. The task analysis required more direct contact with all critical behaviors. It also required that these behaviors be displayed publicly to show that they engaged in them.

These additional elements allowed teachers to make fewer inferences and more objective and concrete observations regarding student performance. For example, when giving a student a problem to solve, students would often solve the problem they think they received instead of the problem they received. A good algorithm will include a list of crucial information. If a student developed an algorithm that solved a problem beautifully and efficiently but did not address the assigned problem, it is wrong.

Additionally, the teacher identified more effectively where the learner went wrong and addressed the issue more directly. See appendix A for task analysis.

Interobserver Agreement

Researchers collected interobserver agreement for a minimum of 35% of sessions with a goal of 80% or better agreement to evaluate data collection consistency across trials. Researchers used the mean score per trial to calculate interobserver agreement during sessions. Researchers calculated agreement by taking the smaller scored number of steps completed divided by the larger scored number of steps completed, then multiplied by 100% to yield a percentage. For example, if one observer scored a 5 and the second observer scored a 4, the agreement would be 4/5, or 80%. If agreement dropped below 80%, researchers would assess whether the discrepancy was related to procedural definitions or training on those definitions. Actual instances of agreement below 80% were exceedingly rare, but those identified were due to ambiguous participant solutions. When ambiguous solutions occurred, scorers scored the event as is. Then, researchers conducted additional training so future scoring on similar events would be consistent.

Experimental Design

The overall experiment included 3 information-gathering components. The first component was pre-experimental. The pre-experimental component included an intake survey for participants so that researchers could ascertain the skill history of participants and any other confounding information relating to a participant's history. The experimental stage comprised the second component. The experimental component was a non-concurrent multiple-baseline across participants included up to 5 different phases across 3 different groups of participants (Kazdin, 1982). The final component was a social validity survey. The social validity survey collected feedback from participants regarding their impressions of the study and its methods.

Pre-Experimental

The pre-experimental portion of the study included an intake survey including sixteen questions divided into three distinct parts. The first section involved personal information and recorded participant name, age, and email. The second section was related to experience and included 7 questions. The experience-related questions asked participants to provide information on if they had engaged with a particular technology or skill and how much experience they had doing so. The final section was related to education and included 6 questions. The education-related section had 3 types of questions. The first type was related to the highest level of education attained by participants. The second type was related to what a participants' major was or, in the case of a changed major or previous higher education, what it had been previously. See Tables 1 and 2 for survey information.

Experimental Stage

The experimental design was a non-concurrent multiple-baseline across participants (Kazdin, 1982). A non-concurrent multiple-baseline allows one participant to transition to the next phase while remaining participants maintain their current phase. Under ideal circumstances, this experimental design may demonstrate or suggest that behavior change occurs due to the independent variable's implementation and not because of some other extraneous variable. This relationship is valid as long as the performance of those not receiving the independent variable does not change until they receive the independent variable.

The researchers chose this design because the independent variables involved teaching and do not allow for the independent variable's withdrawal or reversal once implemented. This design also allows for internal validity through intersubject comparison and external validity by accounting for a more significant number of participants. Finally, this design had the benefits of being flexible regarding scheduling as participants were university students. Therefore, it allowed as many interested students as possible to participate and benefit from the study's positive outcomes (Harvey et al., 2004).

Group 1. Researchers divided participants into 3 uneven groups. The first group included participants who participated from the baseline phase through the Task Analysis + Behavioral Skills Training phase. Group 1 participants were the only participants able to accrue all 5% of extra credit from study participation due to contact with all 5 experimental phases. Participants in Group 1 experienced experimental phases in the following order: baseline, Technical Definition, In-Class Lecture, Task Analysis, and finally, the combined Task Analysis + Behavioral Skills Training phase.

Group 2. The second group included participants recruited later in the semester after the experiment had begun. At this stage, the course instructor had already presented the In-Class Lecture phase. Due to this, participants in this group could only accrue up to 3% extra credit through study participation. The In-Class Lecture phase involved a lecture on writing algorithms as part of the regular CSE1001 Fundamentals of Software

Development 1 course. It, therefore, was administered to all the students and unable to be withdrawn afterward. Due to this, there are some differences in the order of phases. Participants in Group 2 experienced the following phases: In-Class Lecture, Technical Definition, Task Analysis, and Task Analysis + Behavioral Skills Training.

Group 3. The third group included one participant and acted as a control group. The participant in this group was only able to accrue 1% extra credit through study participation. The participant in the control group only experienced probes. The probes were major assignments in the CSE 10001 course. The researcher conducted these probes with Group 3 during the Task Analysis phase and the combined Task Analysis + Behavioral Skills Training phase. The control group allowed researchers to compare a participant who took the course without experimental conditions and participants who experienced experimental conditions.

Social Validity

The social validity survey included 20 Likert scale questions, four free answer questions, and one yes/no question. The free answer question asked if participants had any other feedback regarding how researchers conducted the study. The yes/no question was related to if participants wanted researchers to contact them regarding the final study results. The Likert scale questions asked participants about different elements of the study. The Likert items were: *strongly disagree* (1), *disagree* (2), *neutral* (3), *agree* (4), *strongly agree* (5). The survey had four subscales: Goals, Methods, Results, and Procedures. First, the Goals subscale was composed of two questions, 14 and 15, related to study goals and outcomes. Second, the Methods subscale comprised seven questions which included 1, 6, 7, 11, 16, 18, and 19, and were related to the methods implemented during the study. Third, the Results subscale comprised seven questions that included 4, 8, 9, 10, 12, 13, and 17 and were related to the study's results. Lastly, the Procedures subscale comprised four questions which included 2, 3, 5, and 20, and were related to study procedures.

Procedure

All participants completed an intake survey before the experimental phases and a social validity survey after the experimental phases. Group 1 experienced 5 distinct phases, including baseline, technical definition, In-Class Lecture, Task Analysis, and Task Analysis + Behavioral Skills Training. Group 2 experienced 4 distinct phases, including In-Class Lecture, Technical Definition, Task Analysis, and Task Analysis + Behavioral Skills Training. Group 3 did not experience any phases as the control. Group 3 only had exposure to 3 probes based on the major assignments of the CSE1001 Fundamentals of Software Development 1 course. The first probe occurred during the Task Analysis phase. The other 2 probes occurred during the Task Analysis + Behavioral Skills Training Phase.

Pre-Experimental

Before the experimental phases began, participants completed a sixteen-question intake survey that gathered basic information about their background, general experience, education, and current skills. The intake survey was for later analysis and made it possible for researchers to compare and contrast the experimental phases' effects with participants' individual learning histories. Making this comparison was vital to understanding the final results because it was impossible to fully account for extraneous variables due to learning history during baseline and training phases.

Baseline

Group 1. Group 1 was the only group to experience a baseline phase. Scores gathered during baseline represent participants' performance levels and identified target behaviors present before training and the CSE 1001 Fundamentals to Software 1 course. The data gathered also establish pre-independent variable performance levels. During this phase, participants were given a page from the work problem packet. Researchers then read the problem from the work problem packet out loud for participants. After reading out the problem, researchers gave participants the instruction, "Write an algorithm to solve this problem."

Group 2. Group 2 did not experience a baseline phase. The lack of a baseline phase was due to Group 2 participants joining the study and beginning trials after the In-Class Lecture phase. Therefore, participants would have already had exposure to one of the four independent variables under examination in the study. However, the In-Class Lecture phase was one of the non-behavior analytic independent variables and designed to evaluate traditional instruction methods. Due to this, researchers treated the In-Class Lecture phase as a form of control for participants in Group 2. Instructions given during this phase were identical to those given to Group 1.

Independent Variables

Throughout the study, there were four independent variables: the Technical Definition, In-Class Lecture, Task Analysis, and a package of Task Analysis +

Behavioral Skills Training. The first two independent variables are directly related to typical course instruction. The last two independent variables are related to behavior analytic methods. Each phase continued for at least three trials until data were stable with no greater than a 20% difference along the data path. A variance of 20% difference allowed for a one-point variance in participant scores. Group 1 experienced all phases, including baseline, whereas Group 2 experienced In-Class Lecture and no baseline along with the remaining phases. During the independent variable, researchers gave participants session materials and instructions as described during the baseline phases above with the addition of materials or instruction relevant to each phase.

Technical Definition. Researchers developed the Technical Definition based on the textbook and course materials for CSE1001 Fundamentals of Software Development 1. The Technical Definition represents a typical textbook definition for an algorithm (Crawford, 2019; Hanly et al., 2016). In addition to the materials and instructions described for baseline, participants received a document that included the following: “An algorithm is a set of complete instructions or steps taken to solve a problem. An algorithm is clear and precise, defines inputs and outputs, and has established and measurable parameters.” This intervention's inclusion was because textbook instruction (e.g., textbooks, prompts, etc.) is standard in college courses. Therefore, it was essential to assess the effects such instructional tools would have on individual performance.

In-Class Lecture. This independent variable was the lecture the course instructor gave on algorithm writing during the normal progression of the CSE1001 Fundamentals of Software Development 1 course. Active participants entered this phase regardless of their current progress or placement during the experiment. This independent variable's introduction was a fixed and practically unalterable element of the CSE1001 Fundamentals of Software Development 1 course. Therefore, participant progression into this experimental phase was unavoidable. After the In-Class Lecture, participants completed trials similar to that described in Baseline. In-Class Lecture was included as an independent variable because it represents standard college course instruction. Additionally, researchers included In-Class Lecture because it represented the instruction students typically received on algorithm writing specifically.

Task Analysis. Researchers and the course instructor developed the Task Analysis as a checklist of critical steps that lead to the development of a complete algorithm. The task analysis included the following five components: (1) key information, (2) inputs and outputs, (3) organization, (4) methods used, and (5) sub-problems. The instructions for each step were relatively limited in detail. During sessions, researchers gave participants the task analysis with no additional instructions beyond what researchers delivered during baseline trials. Researchers included this independent variable to evaluate the effectiveness of a task analysis on algorithm writing skills. See appendix A for the task analysis.

Task Analysis + Behavioral Skills Training. This final independent variable was a packaged intervention. During this phase, the task analysis portion of the package was identical to the Task Analysis phase. The Task Analysis was available for participants to reference during each trial. The reason for the package was because it was impossible to withdraw the task analysis effects entirely. The first trial always included all four components of behavioral skills training; instruction, modeling, rehearsal, and feedback. In contrast, subsequent trials only included further instruction and/or modeling as part of feedback when needed or requested by participants. Participants repeated rehearsal and feedback until they reached a score of 80% correct or greater for three consecutive trials. To avoid practice effects, researchers presented a new problem during each iteration of rehearsal.

Instruction. The first component of behavioral skills training was instruction. Elements of instruction intersected with information covered during previous phases, including the Technical Definition, In-Class Lecture, and Task Analysis phases. However, instruction was the only element of behavioral skills training that overlapped with previous phases. During sessions, researchers reviewed the task analysis from the previous phase. Researchers also provided further details regarding the definition of an algorithm as well as an explanation and rationale for their development and implementation.

Modeling. The second component of behavioral skills training was modeling. After the instruction portion was complete, the researcher modeled correct performance. Researchers did this by selecting and completing one of the practice problems that the participant had already completed during an earlier phase. The researchers modeled the practice problem and explained how each step taken related to the task analysis. Instruction and modeling were completed at least once per participant. Researchers repeated this step dependent on participant questions and performance.

Rehearsal. The third component of behavioral skills training was rehearsal or practice. Once the instruction and modeling steps were complete, participants would practice writing an algorithm for a programming problem from the work packet. Researchers expected performance to improve during the first trial of rehearsal after receiving instruction and modeling components in contrast to performance during the Task Analysis alone phase. However, researchers also expected that performance would not reach peak levels until participants could contact feedback, the final component of behavioral skills training.

Feedback. The fourth and final component of behavioral skills training was feedback. During each trial, after the completion of rehearsal, the researcher would review the algorithm participants wrote and grade it based on the criteria covered in the task analysis. Participants received praise for each component completed correctly. Researchers highlighted any components that were missing or incorrect. Researchers then explained to participants how the algorithm could be improved and/or completed. These explanations included examples based on the particular problem. Researchers concluded the feedback by asking participants if they had any questions.

Generalization & Maintenance

Researchers based the generalization and maintenance probes on the three major assignments for the CSE1001 Fundamentals of Software Development 1 course. A Generalization only probe occurred at the end of the Task Analysis phase, and 2 Generalization + Maintenance probes occurred at the end of the Task Analysis + Behavioral Skills Training phase. These assignments required students to write a significantly more complex program than other programs they received as part of regular trials. These assignments were also spaced two to three weeks apart across the length of the semester. Therefore, the assignments probed the generability and maintainability of intervention.

Social Validity

After participants completed the experimental phases, researchers asked participants to complete a social validity survey. This survey gathered information on how the participants felt about the experiment and how participants would rate the value of the experiment's methods. Gathering social validity information was important because of the potential applications to computer science and college instruction. For example, the survey information could change the results or methods' value if a participant reported that they found the methods to be aversive. Aversive methods could lead to a lack of adoption in the future. Alternatively, suppose a participant felt that the results had significant value. In that case, even if some methods were aversive, they may be adopted quickly. Overall, participant feedback could significantly impact future changes and improvements that could influence practical application and adoption into the mainstream.

Chapter 3: Results - Experiment 1

Participants

Experiment 1 consisted of 375 trials conducted across both groups and all 21 participants, with all but one participant completing the study. Researchers conducted each phase for a minimum of three trials with an average of three to four trials per participant. Group 1 consisted of 275 trials across 14 participants, with one participant leaving the study before completion. Group 2 consisted of 97 trials across 6 participants, and Group 3 consisted of 3 trials across one participant. Participants completed between 3 and 5 trials on average per phase, except for the control group, which consisted entirely of one-trial probes. See Tables 1 and 2 for a summary of participants, demographics, and background information for Groups 1, 2.

Interobserver Agreement

Group 1. Researchers collected interobserver agreements for 100% of trials across all phases and participants. Interobserver agreement ranged from 80% to 100% agreement and an average agreement of 99.30%, except for a single score of 60% agreement. Individual average agreement ranged between 92% and 100%. The overall average agreement across phases was 99.31%, with a range of 97.05% to 100%. The overall average agreement was 99.29% across participants, with a range of 98.33% to 100%. See Table 3 for a summary of interobserver agreement data for Group 1.

Group 2. Researchers collected interobserver agreements for 100% of trials across phases and participants with a range of 80% to 100% agreement. Individual averaged agreement ranged between 80% and 100%. Across both axes, the average agreement of interobserver agreement was 97.60%. The overall average agreement across phases was 97.60%, with a range of 93.33% to 100%. The overall average agreement across participants was 97.60%, with a range of 93.33% to 100%. See Table 4 for a summary of interobserver agreement data for Group 2.

Group 3. Group 3 was relatively small and consisted of three specially-targeted probes across a single participant. Researchers collected interobserver agreements for 100% of probes with an average of 93.33% agreement and a range of 80% to 100% agreement.

Participant Scores

Group 1. During baseline, 93% of participants (13 of 14) scored a zero, and one participant scored 20. During the Technical Definition phase, 86% of participants (12 of 14) maintained a score of zero. During the In-Class Lecture phase, 71% of participants (10 of 14) maintained a score of zero. During both phases, participants who increased their score increased to a range of 40-60. There was only a single lecture on algorithms; however, participants continued receiving lectures throughout the course during the study, with about three to nine lectures occurring per phase.

In the Task Analysis phase, 57% of participants (8 of 14) increased their scores to a range of 40-80, with 3 participants meeting the mastery criterion. In comparison, the remaining 43% of participants (6 of 14) maintained a score of zero. At the end of this phase, participant performance was probed based on the second major assignment from the CSE1001 Fundamentals of Software Development 1 course. During this probe, 10 participants maintained scores consistent with those obtained during regular trials. These three participants' scores decreased, and one increased.

During the final phase with the introduction of Task Analysis + Behavioral Skills Training, all but one participant (13 of 14) met the mastery criteria, with 86% of participants (10 of 14) reaching a score of 100. The one participant who did not meet mastery withdrew before data collection was complete. However, based on the data available, this participant showed an increasing trend in their data with a final data point at a score of 100, which meets the mastery criteria. However, there was insufficient data at the mastery criteria to assess stability.

After the final phase, the researchers conducted two probes based on the third and fourth major assignments from the CSE1001 Fundamentals of Software Development 1 course. Two more participants withdrew from the study during these probes, leaving 11 of the original 14 participants to complete the probes. Eight of the remaining 11 participants maintained mastery, with six of the eight maintaining scores of 100 for both probes.

Three participants, Alistair, Brogan, and Dace, produced scores that were inconsistent with what researchers had previously observed during teaching trials. They scored significantly below mastery criteria after consistently meeting the mastery criteria previously. The first participant, Alistair, showed a decrease to just below the mastery criterion. Alistair scored a 60 for the first of the two probes but then returned to a score of 100 for the second and final probe. Brogan displayed a similar decrease to just below the mastery criterion with a score of 60 but across both probes without a return to previous performance.

Participant Dace was an outlier as his scores decreased from 100 during teaching trials to zero for both probes. This contrast shows the most prominent negative change in scores from previous performance across any other participant and phase during the experiment. See Table 5 and Figure 1 for a summary of participant scores for Experiment 1, Group 1.

Group 2. During the In-Class Lecture, 67% of participants (4 of 6) scored zero. The remaining two participants scored a twenty and forty, respectively. With the introduction of the Technical Definition, a strict majority of 50% of participants (3 of 6) maintained a score of zero. Maintaining scores of zero includes the 67% of participants (4 of 6) whose scores were at the same level from baseline. The remaining 33% of participants (2 of 6) increased their score, with a high score of 40.

Upon introducing the task analysis, participants' scores distributed with 33% at 0, 17% at 20, 33% at 40, and 17% at 60, with no participants having reached the mastery criterion. However, two participants did increase their scores from the previous phase, with the highest being a score of 60. Under this phase, 50% of participants (3 of 6) scores did not change, and one participant's scores decreased from the previous phase. At the end of this phase, participants were probed based on the second major assignment from the CSE1001 Fundamentals of Software Development 1 course. During this probe, 50% of participants maintained scores consistent with those obtained during regular trials. Two participants' scores decreased, and one increased.

During the final phase with the introduction of Task Analysis + Behavioral Skills Training, all participants' scores increased and met the mastery criterion at 80 or greater. One participant met mastery at 80, and the remainder met mastery at 100. During this phase, 100% of participants met mastery after just one trial. At the end of the phase, two more probes were conducted based on the third and fourth major assignments from the CSE1001 Fundamentals of Software Development 1 course. For both probes, 50% of participants (3 of 6) maintained scores at 100. One participant maintained a score of 80 for one probe and decreased to a score of 60 on the final probe, just below the mastery criterion. Finally, 33% of participants (2 of 6) scores decreased to 60 for both probes. See Table 6 and Figure 2 for a summary of participant scores for Experiment 1, Group 2.

Group 3. Group 3 was a control group with one participant named Sebastian. Researchers evaluated Sebastian using probes based on the second, third, and fourth major assignments from the CSE1001 Fundamentals of Software Development 1 course. When compared to Group 1 for the second assignment, Sebastian scored a 40. This score was higher than 43% of participants (6 of 14) but lower than 43% of participants (6 of 14). In contrast, only 14% of participants (2 of 14) scoring at the same level for this probe. Compared to Group 2 for the second assignment, Sebastian scored higher than 67% of participants (4 of 6) and lower than 33% of participants (2 of 6). None of the participants scored at the same level for this probe.

When compared to Group 1 for the third assignment, Sebastian scored a 40. This score was lower than all but one participant. This participant was named Dace. Dace was identified earlier as an outlier for scoring a zero during generalization and maintenance probes. They had previously demonstrated a consistent score of 100 during teaching and therefore may not make a good comparison.

A total of 91% of participants (10 of 11) scored above Sebastian. Of participants, 73% (8 of 11) scored at or above the mastery criterion, and 18% of participants (2 of 11) scoring just above Sebastian. For Group 2 of the third assignment, Sebastian did not score higher than any other participants, with 67% of participants (4 of 6) scoring at or above the mastery criterion. 33% of participants (2 of 6) scored just above Sebastian but just below the mastery criterion at a score of 60.

Compared to Group 1 for the fourth assignment, Sebastians scored a 20, decreasing their previous score by 40. Again, this was higher than one participant, the outlier Dace. For the remainder of the participants, 91% scored above Sebastian, with 82% of participants scoring at or above mastery criterion, with just one participant scoring below mastery criterion with a score of 60. For Group 2 of the third assignment, Sebastian again did not score higher than any other participants. Of participants, 50% (3 of 6) scored at or above the mastery criterion. Half of the participants (3 of 6) scored just below the mastery criterion at a score of 60. See Figure 3 for participant scores for Experiment 1, Group 3.

Social Validity

At the end of the study, 11 of the 21 participants completed the social validity survey. Across the 20 questions, the average score was 3.81, with a mode of 5 and a range of 1 to 5. Subscale Goals had an average score of 4, a mode of 5, and a range of 1 to 5. Subscale Methods had an average score of 3.79, with a mode of 5 and a range of 1 to 5. Next, Subscale Results had an average score of 3.79, with a mode of 5 and a range of 1 to 5. Finally, Subscale Procedures had an average score of 3.8, with a mode of 5 and a range of 1 to 5. See Table 7 for a summary of social validity scores.

Chapter 4: Method - Experiment 2

Participants

Participants included eight undergraduate university students. Recruitment of participants occurred from within the undergraduate CSE1001 Fundamentals of Software Development 1 course. All but one participant were men, and all but one participant was between 18 and 20-years-old. Additionally, all but one participant indicated that they had at least one month of computer programming experience. Researchers placed participants into one of two groups based on selection criteria based on current course progression at the time of recruitment during the semester. Group 1 consisted of 3 participants in total, and Group 2 consisted of 5 participants in total. See Tables 8 and 9 for a summary of participants, demographics, and background information for Groups 1 and 2.

Recruitment & Extra Credit

Recruitment procedures and criteria during Experiment 2 were almost identical to those in Experiment 1, except for the participant groups' formation and extra credit distribution. The researchers originally planned for only one experimental group. However, due to circumstances relating to the pace of the course, that was not possible. Therefore, researchers implemented Groups 1 and 2 as detailed in Experiment 1. The distribution of extra credit differed from Experiment 1. The entire 5% extra credit was available to all participants, given that they completed the study once they began.

Setting & Materials

The setting during Experiment 2 was identical to the setting for Experiment 1. Session materials included a work-problem data collection packet, additional information documents, and a pen, as in Experiment 1. Similar to Experiment 1, the work problem packet included a pool of 25 unique problems, which also served as data collection. However, the materials in Experiment 2 had several differences from those in Experiment 1. First, the datasheet and work packet were combined. Second, researchers also redeveloped the packet's work-problems to include the phrase, "Write an algorithm for a..." before describing each problem. Lastly, researchers added a section of each sheet to facilitate the collection of treatment integrity verification data. The alterations to the materials improved the clarity of the instructions and improved resource management. Previously, during Experiment 1, researchers would have to engage with participants between trials to provide instructions for the subsequent trial. This approach required a considerable amount of time. It was also largely unnecessary as long as researchers closely monitored progress and made phase adjustments as necessary.

Dependent Variable

The dependent variable during Experiment 2 was identical to the dependent variable for Experiment 1.

Interobserver Agreement, Treatment Integrity, & Treatment Integrity Verification.

Interobserver Agreement. Procedures for participant scores during Experiment 2 were identical to the interobserver agreement procedures for participant scores in Experiment 1.

Treatment Integrity. In Experiment 2, researchers included Treatment Integrity measures. These measures involved collecting data on which independent variable was present and active during each trial. Participants partially conducted the method for collecting treatment integrity data. Participants collected Treatment Integrity data via a section on the data collection sheet that provided various options. Options included: N/A, A through E, and feedback that the participants could circle or cross out to indicate what supplemental documents or information they had received. Each of the independent variables had a corresponding letter. The baseline corresponded to N/A, and all other independent variables except feedback corresponded to one of the additional information documents. To prevent participants from making guesses about their performance or progress in the study, distractors and non-active options were present on the datasheet. The distractors gave the appearance of randomness and non-linear progression throughout the study. See appendix B for an example of the datasheet with treatment integrity collection elements.

Treatment Integrity Verification. Once researchers collected an initial treatment integrity measure, the researcher and a research assistant later verified it. This measure of interobserver agreement of treatment integrity was called Treatment Integrity Verification. Researchers used treatment integrity verification to discriminate this measure against the traditional interobserver agreement of participant scores. Researchers took treatment integrity verification for a minimum of 35% of sessions at 80% or better. Researchers took treatment integrity verification at this rate to ensure the consistency of treatment across trials and to ensure the consistency of data taken on that treatment. Researchers calculated treatment integrity verification identically to interobserver agreement. If instances of drift in agreement occurred, researchers would assess the data to identify if it was related to procedural definitions or training on those definitions.

Experimental Design

The pre-experimental, social validity, and experimental stages during Experiment 2 were almost identical to those found in Experiment 1. However, in Experiment 2 there were only two participant groups and no control group.

Procedure

The general procedure for Experiment 2 was identical to Experiment 2, save for minor procedural modifications in baseline, the task analysis, and the number of generalization and maintenance probes.

Baseline

The baseline during Experiment 2 was almost identical to Experiment 1. The only difference to baseline was that researchers gave almost no vocal instructions to participants during trials. Researchers would only indicate to participants how many work problems to complete based on individual progress during trials. Researchers would monitor individual participant progress during trials to determine whether a participant could advance to the experiment's next phase. Instructions regarding trials and work problems were primarily delivered textually to participants through the work-problem data collection packet.

Independent Variable

The independent variables during Experiment 2 were almost identical to those in Experiment 1, except for the task analysis steps. The task analysis in Experiment 2 included more detail and a more thorough breakdown of each step than in Experiment 1. Experiment 1 included five steps, whereas Experiment 2 broke each step down to include two sub-sections. The breakdown resulted in 10 individual components of the task analysis. Dividing each step and making them more in-depth allowed for a more precise evaluation of participant performance. Scores differed by 10 points each per correct or incorrect answer during Experiment 2 compared to a 20 point difference during Experiment 1. See appendix C for updated task analysis.

Generalization & Maintenance

Generalization and Maintenance procedures during Experiment 2 were identical to Generalization and Maintenance in Experiment 1. However, the second and third generalization and maintenance probes did not occur due to COVID-19 related issues.

Chapter 5: Results - Experiment 2

Participants

During Experiment 2 there were 106 trials conducted across 8 participants, with only three participants completing the study. Researchers conducted each phase for a minimum of three trials with an average of three to four trials per participant. Experiment 2 included 49 trials across 3 participants for Group 1, with one participant leaving the study before completion. Group 2 included 57 trials across 5 participants, with only one participant completing the study from this group. Participants completed between 3 and 5 trials on average per phase. See Tables 8 and 9 for a summary of participants, demographics, and background information for Groups 1 and 2.

Interobserver Agreement

Group 1. Researchers collected interobserver agreement for 100% of trials across phases and participants with a range of 90% to 100% agreement with an average agreement of 99.26%. Individual averaged agreement ranged between 90% and 100%. The overall average agreement across phases was 99.16%, with a range of 95% to 100%. Additionally, the overall average agreement across participants was 99.44%, with a range of 98.33% to 100%. See table 10 for a summary of interobserver agreement data for Group 1.

Group 2. Researchers collected interobserver agreement for 100% of trials across phases and participants with a range of 80% to 100% agreement. Individual averaged agreement ranged between 90% and 100%. Across both axes, the average agreement of interobserver agreement was 97.94%. Overall average agreement across phases was

98.11%, with a range of 97.33% to 100%. Overall average agreement across participants was 97.78%, with a range of 96.13% to 100%. See table 11 for a summary of interobserver agreement data for Group 2.

Treatment Integrity Verification

Group 1. Researchers collected treatment integrity verification for 100% of trials across phases and participants with an average agreement of 100%. See table 10 for a summary of treatment integrity verification data for Group 1.

Group 2. Researchers collected treatment integrity verification for 92.17% of trials across phases and participants with an average agreement of 100%. The average percentage of treatment integrity verification taken across phases was 93.80%, with a range of 86.67% to 100%. The average percentage of treatment integrity verification taken across participants was 90.55%, ranging from 72.73% to 100%. There was one outlier where treatment integrity verification was absent during the baseline phase for one participant. See table 11 for a summary of treatment integrity verification data for Group 2.

Participant Scores

Group 1. Across Baseline, Technical Definition, In-Class Lecture, and Task Analysis, all participants maintained a score of zero. At the end of the Task Analysis phase, participants were probed based on the second major assignment from the CSE1001 Fundamentals of Software Development 1 course. During this probe, all but one participant maintained scores of zero, consistent with those obtained during regular trials across previous phases. The outlier increased their score to 60 for this probe but did not meet the mastery criteria.

During the final phase with the introduction of Task Analysis + Behavioral Skills Training, there was attrition of one participant due to the COVID-19 pandemic. All of the remaining participants demonstrated an increase in scores and met the mastery criterion. However, despite having met the mastery criterion, scores were slightly more variable during this phase than scores from participants from the first experiment during their final independent variable phase. After this phase, researchers planned probes based on the third and fourth assignments for the CSE1001 Fundamentals of Software Development 1 course. However, they were unable to be conducted due to the COVID-19 pandemic. See Table 12 and Figure 4 for a summary of participant scores for Experiment 2, Group 1.

Group 2. During the In-Class Lecture, 40% of participants (2 of 5) scored a zero, with the remainder distributed between scores of 10 and 50. This distribution persisted with no participants scoring over 50 for the Technical Definition phases. No participants scored over 60 for the Task Analysis phases. Of participants, 60% (3 of 5) maintained a consistent level across the three phases. By comparison, one participant showed minor variations. Of participants, 40% (2 of 5) showed an increase in their score during the Task Analysis phase. The first participant showed a minimal but stable increase of 10 from the previous with an end score of 40. Next, the second showed a slightly more significant increase of 30 from the previous with an end score of 60. No participants had reached mastery criteria at this stage of the experiment. At the end of the Task Analysis phase, participants were probed based on the second major assignment from the

CSE1001 Fundamentals of Software Development 1 course. During this probe, 60% of participants (3 of 5) showed an increase in score by 20. However, this was not enough for any of these participants to meet mastery. The remaining 40% of participants (2 of 5) maintained the same score as shown during previous trials.

During the final phase with the introduction of Task Analysis + Behavioral Skills Training, there was attrition of four participants due to the COVID-19 pandemic leaving only one participant in this group, participant Daveth. Participant Daveth saw a significant increase in score and met mastery criteria with a score of 100. Unlike other participants during this phase, this participant met mastery at 100 on the first trial instead of a gradual increase to mastery. The high initial scores meant that only the instruction, modeling, and rehearsal components of behavioral skills training were active at the time of masty. Researchers delivered feedback regardless, however with this participant's score at the maximum level, researchers could not evaluate the effects of feedback. After this phase, researchers planned probes based on the third and fourth assignments for the CSE1001 Fundamentals of Software Development 1 course. However, they were unable to be conducted due to the COVID-19 pandemic. See Table 13 and Figure 5 for a summary of participant scores for Experiment 2, Group 2.

Social Validity

At the end of the study, seven of the eight participants completed the social validity survey. Across the 20 questions, the average score was 4.41, with a mode of 5 and a range of one to 5. Subscale Goals had an average score of 4.43, a mode of 5, and a range of 2 to 5. Subscale Methods had an average score of 4.45, with a mode of 5 and a range of 3 to 5. Next, subscale Results had an average score of 4.43, with a mode of 5 and a range of one to 5. Subscale Procedures had an average score of 4.29, with a mode of 5 and a range of one to 5. See Table 14 for a summary of social validity scores.

Chapter 6: Discussion

Algorithm writing is a critical, fundamental skill for computer scientists, which has only grown in importance as computers become increasingly integrated into every facet of society. Therefore, the purpose of this study was to evaluate the effectiveness of current teaching methods for algorithm writing for undergraduate students in an introductory computer science course. The experimenters then compared different behavior-based approaches to identify best practices for teaching algorithm writing. This study's results may contribute to the future development of assignments and class exercises that can be made available to a wide variety of classes without significant training or modification.

Participants

Participant recruitment occurred within the undergraduate CSE 1001 Fundamentals of Software Development 1 course. According to the intake survey, all participants had an established academic background taking coursework of similar difficulty to CSE1001 Fundamentals of Software Development 1. They also had a history of engaging in problem-solving tasks and were at least a little familiar with the subject matter. Outliers included age differences. Most participants were between the ages of 18 and 22. In contrast, 14% of participants (4 of 29) were either younger or older than this age range. There were also very few participants who were women (17%). However, this age and gender-identity difference did not appear to have any effect on participant performance.

In terms of ethnicity, participants were much more diverse than typical populations seen within computer science. Out of both experiments, 38% of participants (8 of 21) from Experiment 1 and 62.5% of participants (5 of 8) from Experiment 2 were White men. These participants stood out as significant outliers as typical computer science demographics are around 70% White men (Bureau of Labor Statistics, 2019; Data USA n.d.). The participant population could have been more diverse. However, these participants' results indicate more representative findings of the broader population than would otherwise be available.

Recruitment & Baseline

During recruitment of participants, extra credit of up to 5% to their final grade was available for participation. To address concerns that the extra credit may have a coercive effect on student participation, the professor offered an additional assignment for those who did not want to participate. However, this measure may not have been enough to address the full impact of extra credit on participants. The extra credit could have led to selection bias amongst participants. For example, students with significant programming, algorithm writing, or other computer science related skill deficits may have been more inclined to participate. These participants might have expected to perform worse in the CSE1001 Fundamentals of Software Development 1 course if they had otherwise not participated. Additionally, the bias may lead to an overestimation of skill deficits within similar student populations, impacting how incoming students are perceived and assessed.

An additional factor that may have had an impact on recruitment was the timing of the study. Both Experiments 1 and 2 were conducted in conjunction with the CSE1001 Fundamentals of Software Development 1 course. Several elements of the study, including an independent variable, were directly linked to the course's pace. Therefore, participants only had a few weeks from the start of the semester to decide if they wanted to participate. One of the ways this could have impacted recruitment is that students may have been hesitant to participate due to the time commitment. Given the many commitments a student has, it may be unclear how much time a student has to commit to participation. Another impact was that students do not know how they will perform in the class at the start of the semester. Without foreknowledge of academic performance, the opportunity for extra credit initially may be less valuable. However, this may mean that there was less potential for recruitment bias for those students who chose to participate before extra credit could become a significant motivating factor.

Researchers placed participants who joined during the initial recruitment period into Group 1 during both experiments. The baseline results for Group 1 from Experiments 1 and 2 suggest that universities cannot expect incoming students to have algorithm writing skills. Researchers placed participants who joined after the In-Class Lecture into Group 2 for both Experiment 1 and 2. While random assignment into groups would have been preferable, it was not possible due to recruitment timing. Participants placed into Group 2 are the most likely candidates for bias because of the time in which they joined. However, there was no way to determine if joining late was due to poor academic performance in the course or because of another factor. The difference between

Group 1 and 2 did not appear to impact performance based on the results from overlapping phases. Regardless, late recruitment resulted in participants from Group 2 being unable to undergo baseline.

Technical Definition & In-Class Lecture

Group 1 and 2 received the Technical Definition and In-Class Lecture in the opposite order. Group 1 received the Technical Definition first after baseline, followed next by In-Class Lecture. Group 2 received the In-Class Lecture first in place of baseline followed by the Technical Definition. The difference in order was unavoidable due to the exposure of Group 2 to the In-Class Lecture. However, it allowed for examining whether or not the addition of a Technical Definition would be more beneficial after rather than before receiving the In-Class Lecture.

Results from the Technical Definition and In-Class Lecture phases indicate three things. First, the order in which these interventions occur does not significantly impact performance. Second, access to a textbook containing a technical definition was not enough to occasion algorithm writing skills to occur correctly. Lastly, while students may benefit from a lecture, lecture alone was not enough to occasion correct algorithm writing skills. Therefore, clear comparisons exist between a technical definition, instruction, and task clarification. Materials may provide different levels of information and detail. For example, a task analysis provides a narrow and specific level of detail. An instruction provides a broad and intensive level of detail. In contrast, a textbook was likely to provide a technical definition that consists of a short definition and possibly an illustrative example but not an exhaustive list of steps.

Task Analysis & Behavioral Skills Training.

With the introduction of the next phase, the Task Analysis, performance did not significantly improve. To ensure that the regular tasks participants completed were not perceived as too easy, causing participants to disregard the task analysis, researchers utilized an additional probe. Researchers based the probe on a major course assignment of significantly greater complexity than the tasks participants encountered during regular trials. The added complexity of the probe modified the potential value of the task analysis. However, the probe indicated that the task's complexity had no impact on participants' motivation to use the task analysis or that the task analysis was not effective. Additionally, during Experiment 2, the task analysis was expanded to include more detail. Unfortunately, this had no impact on performance when compared to Experiment 1 across both the regular trials and probe.

Task analyses, checklists, and other similar step-by-step instructional tools are common in the literature but seldom evaluated by themselves without being combined with consequences such as feedback. Consistent with Anderson et al. (1988), a task analysis improved performance but not by a significant amount. Additionally, the Anderson et al. (1988) results were predictive of the results following the addition of behavioral skills training. Consistent and significant improvement across all participants occurred when the final phase, Task Analysis + Behavioral Skills Training, was introduced. However, while instruction and modeling significantly improved performance, performance did not reach 100% correct until after receiving feedback at least once.

Although this study did not conduct a component analysis of behavioral skills training, the results are consistent with a component analysis conducted by Lewon et al. (2019). Lewon et al. (2019) showed gradual performance improvements with each additional component of behavioral skills training. Similarly, a significant performance improvement as a result of instruction and modeling alone was understandable. However, much of the performance improvement in this regard was likely due to the implementation of modeling. The instruction component of behavioral skills training was based directly on the task analysis. Like Speelman et al. (2015), researchers reviewed the task analysis during instruction, including an explanation and rationale. Due to the lack of significant difference in detail between the task analysis and instruction, instruction's impact was likely similar to that of the task analysis phase.

Modeling had a slight difference between Experiment 1 and 2. Experiment 1 included a modeling phase that was in-vivo. In-vivo, in this context, means that while the researcher was a subject matter expert, the implementation of the modeling was not strictly structured. On the other hand, Experiment 2 was more rehearsed and included prepared examples that touched on essential components more explicitly. While performance was high during Experiment 1, there were slightly fewer trials required during Experiment 2. The difference between performance in between Experiment 1 and 2 was minimal. However, these results indicate that better quality modeling leads to better performance improvement overall (Seiverling et al., 2012).

After the implementation of feedback, performance continued to improve across all participants. Participant performance improved to reach the mastery criterion and often reached 100% by the end of the phase. These findings are consistent with other literature and suggest that feedback is a crucial component. Literature suggests that feedback is most effective when combined with other methods; however, researchers could not verify this without a component analysis (Alvero et al., 2001). Additionally, these findings are consistent with research that indicates that performance does not reach its highest point until the implementation of feedback (Lewon et al., 2019).

Generalization & Maintenance

Researchers planned two generalization and maintenance probes to occur several weeks after the Task Analysis + Behavioral Skills Training trials. Due to COVID-19 related complications, these probes only occurred for Experiment 1. These probes were similar to the probe during the Task Analysis phase and based on similar major course assignments. The goal was to determine if the skill would maintain over time and generalize to the more complex task. Most participants maintained their previous performance. However, some participant's performance did decrease. Some of these participants' performance decreased to below the mastery criterion, usually by a small margin. Decreased performance during these probes might be due to three main factors. The first was that the time between teaching trials and the probes resulted in decreased performance. The second was that the complexity of the task made it difficult to generalize the skill. The third was that the probes occurred during the end of the school year, typically a hectic time that could have been a significant distraction.

One participant stood out in particular. Participant Dace was an outlier as his scores decreased from 100 during teaching trials to zero for both probes. This contrast shows the most significant negative change in scores from previous performance across any other participant and phase during the experiment. The performance of Dace was also in contrast to the control group, which received no interventions but completed all three probes during Experiment 1. Consistent with receiving no intervention, the control consistently performed lower than all other participants except Dace during these last two probes.

Dace reverted to performance typically seen during baseline and the first two phases. During these phases, participants would typically write code instead of an algorithm. It was unclear what may have occurred given that Dace had experienced a probe before and performed very well during training trials. However, stimulus control might have weakened during probes because they occurred during the end. The end of the semester typically requires students to complete major projects and sit for final exams. These additional environmental elements may have reduced the value of completing the probes correctly. Additionally, it may have been less response effort to turn in code since the student had to write code to complete the course assignment.

Control & Causality

The control group was only present for Experiment 1 and consisted of only a single participant. The participant was provided with the same instructions as the other participants but was limited to completing only the probe without any additional information. Researchers based the probes on the same major assignments for the CSE1001 Fundamentals of Software Development 1 course that other participants received. While this participant's scores were consistent and low, these scores did not act as a significant control compared to the other groups because there was only one participant. The original expectation of researchers was that there would be more varied interest in participation by students. Therefore, researchers expected that there would be more participants in the control group. Additionally, because researchers did not expose this participant to regular trials outside of the probes, it was not possible to compare the performance of Group 1 and 2 participants and the control for regular trials.

The lack of participants in the control group means that the data may not represent the effects of extraneous variables for most other participants but how these variables specifically affected this one participant. Additional issues arose because of the number of participants in Groups 1 and 2 and how the CSE 1001 Fundamentals of the Software Development 1 course bound the pacing of the study. The number of participants and pacing of the course made it challenging to correctly implement the non-concurrent multiple baseline. The high numbers of participants made it difficult to arrange the experimental phases to demonstrate causal control of each independent variable as researchers implemented them. The result was that there was no significant overlapping data to demonstrate that when one participant changed phase, the other participants' performance was not affected.

Social Validity

Researchers distributed a social validity survey amongst participants after they had completed the last generalization and maintenance probe. Overall, participants rated the study very highly. However, two outliers consistently rated all options negatively. The scoring from one of these participants was incongruent with the general affect and opinion voiced during trials. The participant may have masked their genuine opinion on the study. However, given that the participant stood out and showed a consistent negative score, this participant most likely scored each question opposite as intended by accident.

Overall, feedback from participants indicated that they would have preferred to receive behavioral skills training, or something like it, from the start. This feedback may suggest that behavioral skills training was a highly valid approach in terms of methodology. They further elaborated that it was very frustrating not to get feedback, ask questions, or have the opportunity to know how they are doing. While providing some limited information regarding performance, such as a score, would likely have not had a meaningful influence on performance, it is a form of feedback. It would have introduced additional variables to other phases and made it difficult to evaluate the effects of those phases on their own. More specific requests from participants included access to more examples, non-examples, and contrasts between the two. The apparent value of task analysis + behavioral skills training aside, consistent with Feld and Grofman (1977) and Kim et al. (2019), students again reported that they prefer engagement to didactic instruction. Therefore, the survey indicated that task analysis + behavioral skills training was effective as well as preferred.

COVID-19

Covid-19 was a significant consideration during Experiment 2. During the spring semester, when Experiment 2 occurred, schools and universities worldwide switched to an all-online educational approach due to safety concerns regarding the virus. In addition to these changes, some additional concerns and logistical challenges included communication, health, and the physical location of students. These considerations occurred during the conclusion of the task analysis phase and the start of the task analysis + behavioral skills training phase. While researchers implemented steps to ensure sanitation and social distancing, it became prohibitively difficult to contact participants and conduct research sessions. These obstacles resulted in only a handful of participants being able to complete the final stages of the study. Furthermore, researchers decided to discard the planned generalization and maintenance probes that had previously followed the task analysis + behavioral skills training phases.

Future Directions

Behavior analytic methods like task analysis and behavior skills training have a long successful history of teaching various skills. Behavior analysis also shows significant promise in teaching algorithm writing skills. The importance of algorithm writing skills cannot be understated, especially for new computer scientists. Learning to perform this skill correctly helps establish the foundation for high performance and learning with more complex computer science skills in the future. Therefore, further research in the area of teaching algorithm writing skills is warranted.

One future direction would be to replicate and expand this study to eliminate or refine some of the less desirable elements. For instance, some changes to recruitment and group placement could be beneficial. The implementation of these changes could occur by conducting a study in an independent setting from an active college course. First, this would eliminate the selection bias by removing the extra credit component and may lead to a more neutral recruitment process. Second, it would remove the need for more than one experimental group because researchers would have more control over the intervention introduction. Finally, it would allow more control by researchers over the interventions.

Additionally, another element that researchers could refine would be the use of surveys. For example, researchers could change both the intake and social validity surveys to require fewer questions. To accomplish this, researchers could use more concise and targeted language. Additionally, researchers could implement questions that alternate so that participants cannot select one side of a set of questions for every question. These alterations would be beneficial because they could collect more meaningful information and reduce the amount of time required to complete and analyze the surveys' results. More meaningful information means a better understanding of participant history and feedback.

Another future direction could be to do a more in-depth and focused analysis like a component analysis of task analysis and behavioral skills training and their impact on teaching algorithm writing skills. The results show that the use of task analyses and behavioral skills training leads to beneficial performance improvements. What is not clear from the results is to what extent each component of these interventions impacts results. For example, things like task analyses and instructions provide information on performance which leads to improvement. However, we do not know how much or what kind of information in these components will yield the best results. It may be that the volume or level of details contained in these components can have a significant influence over their effectiveness. See appendix D for a proposal of a future study.

Furthermore, more what constitutes adequate and effective modeling has yet to be determined. It is also possible to say the same regarding feedback. Exploring these elements, especially in teaching a skill like algorithm writing, may significantly improve how these interventions function and improve performance. Finally, understanding how to apply each of the components of task analyses and behavioral skills training with precision can lead to developing a course curriculum that does not require significant alterations to courses that already exist. Better application may result in faster adoption and more immediate results for universities that may reduce faculty demands and improve outcomes for students.

References

Academic Positions. (2018). *PhD, professor, and postdoc salaries in the United States.* https://academicpositions.com/career-advice/phd-professor-and-postdoc-salaries-in-the-united-states

Aherne, C. M., & Beaulieu, L. (2018). Assessing long-term maintenance of staff performance following behavior skills training in a home-based setting. *Behavioral Interventions*, *34*(1), 79-88. https://doi.org/10.1002/bin.1642

Alvero, A. M., Bucklin, B. R., & Austin, J. (2001). An objective review of the effectiveness and essential characteristics of performance feedback in organizational settings (1985-1998). *Journal of Organizational Behavior Management*, *21*(1), 3–29. https://doi.org/10.1300/J075v21n01_02

Anderson, D. C., Crowell. C. R., Hantula, D. A. & Siroky, L. M. (1988). Task clarification and individual performance posting for improving cleaning in a student-managed university bar, *Journal of Organizational Behavior Management, 9*(2), 73-90. https://doi.org/10.1300/J075v09n02_06

Austin, J. L., & Soeda, J. M. (2009). Effective teaching, effective living: A review of behavior analysis for effective teaching by Julie S. Vargas. *Behavior Analysis in Practice*, *2*(2), 63–68. https://doi.org/10.1007/BF03391750

Bacon, D. L., Fulton, B. J., & Malott, R. W. (1983). Improving staff performance through the use of task checklists. *Journal of Organizational Behavior Management*, *4*(3-4), 17-25. https://doi.org/10.1300/J075v04n03_03

Baer, D.M., Wolf, M.M. & Risley, T.R. (1968). Some current dimensions of applied behavior analysis. *Journal of Applied Behavior Analysis, 1*(1), 91-97. https://doi.org/10.1901/jaba.1968.1-91

Baeza-Yates, R. A. (1995). Teaching algorithms. *SIGACT News*, *26*(4), 51-59. https://doi.org/10.1145/219817.219828

Barker, L., Moore, J. W., Olmi, J. D., & Rowsey, K. (2019) A comparison of immediate and post-session feedback with behavioral skills training to improve interview skills in college students. *Journal of Organizational Behavior Management, 39*(3-4), 145-163. https://doi.org/10.1080/01608061.2019.1632240

Beckschi, P., & Doty, M. (2000). Instructional systems design: A little bit of ADDIEtude, please. In G. M. Piskurich, P. Beckschi, & B. Hall (Eds.), *The ASTD handbook of training design and delivery*, (pp. 28–41). McGraw-Hill.

Belisle, J., Rowsey, K. E., & Dixon, M. R. (2016). The use of in situ behavioral skills training to improve staff implementation of the PEAK relational training system. *Journal of Organizational Behavior Management*, *36*(1), 71–79. https://doi.org/10.1080/01608061.2016.1152210

Bureau of Labor and Statistics. (2017). 69.7 percent of 2016 high school graduates enrolled in college in October 2016. https://www.bls.gov/opub/ted/2017/69-point-7-percent-of-2016-high-school-graduates-enrolled-in-college-in-october-2016.htm

Bureau of Labor and Statistics. (2017). *Women in architecture and engineering occupations in 2016.* https://www.bls.gov/opub/ted/2017/women-in-architecture-and-engineering-occupations-in-2016.htm

Bureau of Labor and Statistics. (2019). *College enrollment and work activity of recent high school and college graduates summary.* https://www.bls.gov/news.release/hsgec.nr0.htm

Bureau of Labor and Statistics. (2019). *Computer and information technology occupations.* https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Bureau of Labor and Statistics. (2019). *Labor force statistics from the current population survey.* https://www.bls.gov/cps/cpsaat11.htm

Chiesa, M. & Robertson, A. (2000). Precision teaching and fluency training: Making maths easier for pupils and teachers. *Educational Psychology in Practice, 16*(3), 297-310. https://doi.org/10.1080/713666088

Carrow, J.N., Vladescu, J.C., Reeve, S.A. & Kisamore, A.N. (2020). Back to sleep: Teaching adults to arrange safe infant sleep environments. *Journal of Applied Behavior Analysis.* https://doi.org/10.1002/jaba.681

Cooper, S., & Cunningham, S. (2010). Teaching computer science in context. *ACM Inroads*, *1*(1), 5-8. https://doi.org/10.1145/1721933.1721934

Crawford, H. (2019). *Algorithms and Errors.* Unpublished PowerPoint. Department of Computer Engineering and Sciences, Florida Institute of Technology.

Crowell, C. R., Anderson, D. C., Abel, D. M., & Sergio, J. P. (1988). Task clarification, performance feedback, and social praise: Procedures for improving the customer service of bank tellers. *Journal of Applied Behavior Analysis*, *21*(1), 65–71. https://doi.org/10.1901/jaba.1988.21-65

Daniels, A. C., & Bailey, J. S. (2014). *Performance Management: Changing behavior that drives organizational effectiveness (5th ed.).* Aubrey Daniels International, Inc.

Data USA. (n.d.). Computer, engineering, & science occupations. https://datausa.io/profile/soc/computer-engineering-science-occupations

Day-Watkins, J., Pallathra, A. A., Connell J. E., & Brodkin, E. S. (2018) Behavior skills training with voice-over video modeling, *Journal of Organizational Behavior Management, 38*(2-3), 258-273. https://doi.org/10.1080/01608061.2018.1454871

Denning, P. J. (1989). A debate on teaching computing science. *Communications of the ACM*, *32*(12), 1397-1414. https://doi.org/10.1145/76380.76381

Durgin, A., Mahoney, A., Cox, C., Weetjens, B. J., & Poling, A. (2014). Using task clarification and feedback training to improve staff performance in an East African nongovernmental organization. *Journal of Organizational Behavior Management*, *34*(2), 122-143. https://doi.org/10.1080/01608061.2014.914007

Dogan, R.K., King, M.L., Fischetti, A.T., Lake, C.M., Mathews, T.L., & Warzak, W.J. (2017). Parent-implemented behavioral skills training of social skills. *Journal of Applied Behavior Analysis, 50*(4), 805-818. https://doi.org/10.1002/jaba.411

Feld, S. L., & Grofman, B. (1977). Variation in class size, the class size paradox, and some consequences for students. *Research in Higher Education, 6*, 215–222. https://doi.org/10.1007/BF00991287

Fertig, J. (2012, March 29). How do professors learn to teach (or do they)? *The James G. Martin Center for Academic Renewal.* https://www.jamesgmartin.center/2012/03/how-do-professors-learn-to-teach-or-do-they/

Forišek, M. & Steinová, M. (2012). Metaphors and analogies for teaching algorithms. *SIGCSE '12: Proceedings of the 43rd ACM technical symposium on Computer Science Education.* Association for Computer Machinery. https://doi.org/10.1145/2157136.2157147

Gelles, D. (2019, October 28). Boeing 737 Max: What's happened after the 2 deadly crashes. *The New York Times*. https://www.nytimes.com/interactive/2019/business/boeing-737-crashes.html

Gettinger, M. (1993). Effects of invented spelling and direct instruction on spelling performance of second-grade boys. *Journal of Applied Behavior Analysis, 26*(3), 281-291. https://doi.org/10.1901/jaba.1993.26-281

Hanly, J. R. & Koffman, E. B. (2016). *Problem solving and program design in c*. Pearson.

Harvey, M.T., May, M.E. & Kennedy, C.H. (2004). Nonconcurrent Multiple Baseline Designs and the Evaluation of Educational Systems. *Journal of Behavioral Education, 13*(4), 267–276. https://doi.org/10.1023/B:JOBE.0000044735.51022.5d

Helmreich, R. L., Klinect, J. R., Wilhelm, J. A., & Jones,S. G. (1999). The line/LOS error checklist, Version6.0: A checklist for human factors skills assessment, a log for off-normal events, and a worksheet for cockpit crew error management (Tech. Rep. No. 99-01). Austin: University of Texas, Human Factors Research Project.

Helmreich, R. L., Wilhelm, J. A., Klinect, J. R., &Merritt, A. C. (2001). Culture, error, and crew resource management. In E. Salas, C. A. Bowers, & E.Edens (Eds.), Improving teamwork in organizations(pp. 305–331). Hillsdale, NJ: Erlbaum.Himle, M.B. & Wright, K.A. (2014). Behavioral skills training to improve installation and use of child passenger safety restraints. *Journal of Applied Behavior Analysis, 47*(3), 549-559. https://doi.org/10.1002/jaba.143

Jenkins, S. R. & DiGennaro Reed, F. D. (2016). A parametric analysis of rehearsal opportunities on procedural integrity, *Journal of Organizational Behavior Management, 36(*4), 255-281. https://doi.org/10.1080/01608061.2016.1236057

Kaplan, J. (2020, Jun 9). One of the big three rating agencies sees college enrollment down as much as 20% for colleges this fall. *Business insider.* https://www.businessinsider.com/college-enrollment-decline-could-reach-20-private-schools-hit-harder-2020-6

Kazdin, A. E. (2011). *Single-case research designs (2nd ed.).* Oxford University Press.

Keller F. S. (1968). Good-bye, teacher... *Journal of Applied Behavior Analysis*, *1*(1), 79–89. https://doi.org/10.1901/jaba.1968.1-79

Kienitz, P. (2019, August 27). Most expensive software mistakes. *DCSLSoftware.* https://www.dcslsoftware.com/most-expensive-software-mistakes/

Kim, A. S. N., Shakory, S., Azad, A., Popovic, C., & Park, L. (2019). Understanding the impact of attendance and participation on academic achievement. *Scholarship of Teaching and Learning in Psychology*. Advance online publication. https://doi.org/10.1037/stl0000151

Kolmar, C. (2020, November 6). What is a subject matter expert and what do they do? *Zippia.* https://www.zippia.com/advice/subject-matter-expert/

Kranak, M. P., Shapiro, M. N., Sawyer, M. R., Deochand, N., & Neef, N. A. (2019). Using behavioral skills training to improve graduate students' graphing skills. *Behavior Analysis: Research and Practice*, *19*(3), 247-260. http://doi.org/10.1037/bar0000131

Lewon, M., Webb, E. K., Brotheridge, S. M., Cox, C. & Fast, C.D. (2019). Behavioral skills training in scent detection research: Interactions between trainer and animal behavior. *Journal of Applied Behavior Analysis, 52*(3), 682-700. https://doi.org/10.1002/jaba.566

Lindsley, O. R. (1992). Precision teaching: Discoveries and effects. *Journal of Applied Behavior Analysis, 25*(1), 51-57. https://doi.org/10.1901/jaba.1992.25-51

Mayfield, K. H., & Chase, P. N. (2002). The effects of cumulative practice on mathematics problem solving. *Journal of Applied Behavior Analysis*, *35*(2), 105-123. https://doi.org/10.1901/jaba.2002.35-105

Merbitz, C., Vieitez, D., Merbitz, N. H., & Binder, C. (2004). *Precision teaching: Applications in education and beyond.* In D. J. Moran & R. W. Malott (Eds.), *A Vol. in the educational psychology series. Evidence-based educational methods* (p. 63–78). Elsevier Academic Press. https://doi.org/10.1016/B978-012506041-7/50006-1

Michael, J. A. (1991). A behavioral perspective on college teaching. *The Behavior Analyst*, *14*(2), 229–239. https://doi.org/10.1007/BF03392578

Miltenberger, R. G. (2008). Teaching safety skills to children: Prevention of firearm injury as an exemplar of best practice in assessment, training, and generalization of safety skills. *Behavior Analysis in Practice*, *1*, 30-36. https://doi.org/10.1007/BF03391718

National Center for Educational Statistics. (2011). How is grade point average calculated? https://nces.ed.gov/nationsreportcard/hsts/howgpa.aspx

Nigro-Bruzzi, D. & Sturmey, P. (2010). The effects of behavioral skills training on mand training by staff and unprompted vocal mands by children. *Journal of Applied Behavior Analysis, 43*(4), 757-761. https://doi.org/10.1901/jaba.2010.43-757

Palmer, M. G., Johnson, C. M., & Johnson, D. A. (2015). Objective performance feedback: Is numerical accuracy necessary? *Journal of Organizational Behavior Management*, *35*(3-4), 206-239. https://doi.org/10.1080/01608061.2015.1093059

Parsons, M. B., Rollyson, J. H., & Reid, D. H. (2012). Evidence-based staff training: A guide for practitioners. *Behavior Analysis in Practice*, *5*(2), 2–11. https://doi.org/10.1007/BF03391819

Rantz, W. G., Dickinson, A. M., Sinclair, G. A., & Houten, R. V. (2009). The effect of feedback on the accuracy of checklist completion during instrument flight training. *Journal of Applied Behavior Analysis*, *42*(3), 497–509. https://doi.org/10.1901/jaba.2009.42-497

Resnick, L. B., Wang, M. C., & Kaplan, J. (1973). Task analysis in curriculum design: A hierarchically sequenced introductory mathematics curriculum. *Journal of Applied Behavior Analysis*, *6*(4), 679-709. https://doi.org/10.1901/jaba.1973.6-679

Robbins, J. K. (2011). Problem solving, reasoning, and analytical thinking in a classroom environment. *The Behavior Analyst Today*, *12*(1), 41-48. http://doi.org/10.1037/h0100710

Rosales, R., Stone, K. & Rehfeldt, R. A. (2009). The effects of behavioral skills training on implementation of the picture exchange communication system. *Journal of Applied Behavior Analysis, 42*(3), 541-549. https://doi.org/10.1901/jaba.2009.42-541

Saville, B. K., Pope, D., Truelove, J., & Williams, J. (2012). The relation between GPA and exam performance during interteaching and lecture. *The Behavior Analyst Today*, *13*(3-4), 27-31. http://doi.org/10.1037/h0100728

Saville, B. K., Zinn, T. E., Neef, N. A., Norman, R. V. & Ferreri, S.J. (2006). A comparison of interteaching and lecture in the college classroom. *Journal of Applied Behavior Analysis, 39*(1), 49-61. https://doi.org/10.1901/jaba.2006.42-05

Seiverling, L., Williams, K., Sturmey, P. & Hart, S. (2012). Effects of behavioral skills training on parental treatment of children's food selectivity. *Journal of Applied Behavior Analysis, 45*(1), 197-203. https://doi.org/10.1901/jaba.2012.45-197

Shein, E. (2019). The CS teacher shortage. *Communications of the ACM*, *62*(10), 17-18. https://doi.org/10.1145/3355375

Skinner, B. F. (1968). *The technology of teaching.* New York: Appleton-Century-Crofts.

Speelman, R. C., Whiting, S. W., & Dixon, M. R. (2015). Using behavioral skills training and video rehearsal to teach blackjack skills. *Journal of Applied Behavior Analysis*, *48*(3), 632-642. https://doi.org/10.1002/jaba.225

Stocco, C.S., Thompson, R. H., Hart, J. M. & Soriano, H. L. (2017). Improving the interview skills of college students using behavioral skills training. *Journal of Applied Behavior Analysis, 50*(3), 495-510. https://doi.org/10.1002/jaba.385

Study Portals. (2021, January 14). 5 reasons why university rankings are not perfect. https://www.mastersportal.com/articles/2023/5-reasons-why-university-rankings-are-not-perfect.html

Suberman, R. & Cividini-Motta, C. (2020). Teaching caregivers to implement mand training using speech generating devices. *Journal of Applied Behavior Analysis, 53*(2), 1097-1110. https://doi.org/10.1002/jaba.630

Sulzer-Azaroff, B. (1985). Behavior analysis and education: Crowning achievements and crying needs. University of Massachusetts at Amherst.

Sump, L.A., Richman, D.M., Schaefer, A.M., Grubb, L.M. & Brewer, A.T. (2018). Telehealth and in-person training outcomes for novice discrete trial training therapists. *Journal of Applied Behavior Analysis, 51(*3), 466-481. https://doi.org/10.1002/jaba.461

Swail, W. S. (2016, November 28). So how much does student departure cost your institution? *Educational Policy Institute.* https://theswailletter.com/2016/11/28/so-how-much-does-student-departure-cost-your-institution/

Tai, S.S.M. & Miltenberger, R.G. (2017). Evaluating behavioral skills training to teach safe tackling skills to youth football players. *Journal of Applied Behavior Analysis, 50*(4), 849-855. https://doi.org/10.1002/jaba.412

Tudor, R. M. & Bostow, D.E. (1991). Computer-programmed instruction: The relation of required interaction to practical application. *Journal of Applied Behavior Analysis, 24*(2) 361-368. https://doi.org/10.1901/jaba.1991.24-361

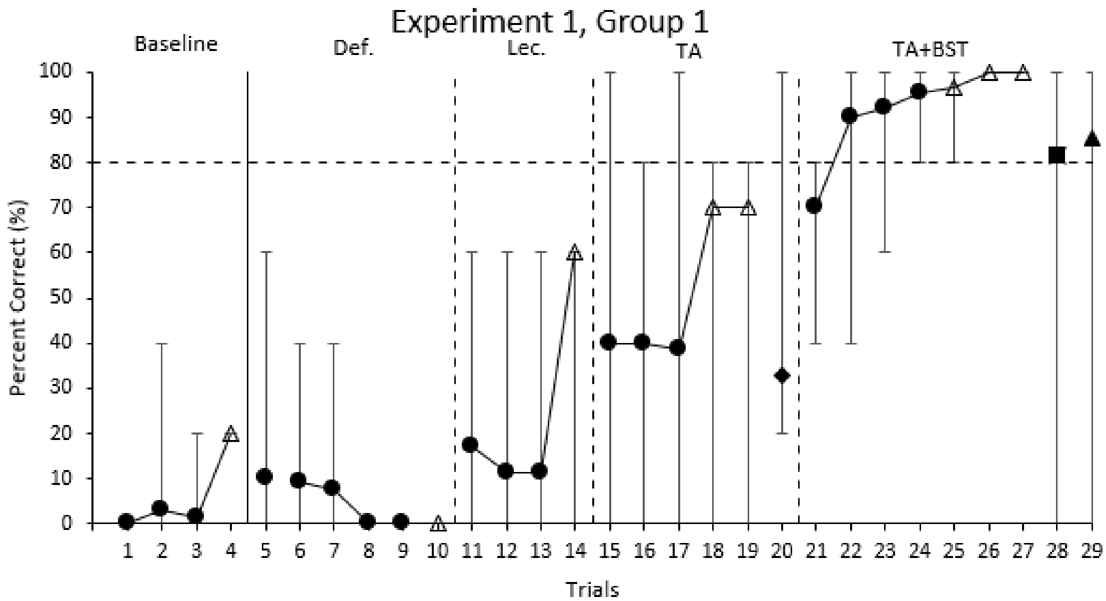
Twyman, J. (2014). Competency-based education: Supporting personalized learning. Center on Innovations in Learning. http://www.centeril.org/connect/resources/Connect_CB_Education_Twyman-2014_11.12.pdf

U.S. Department of Education. (2020, December 12). Accreditation in the United States. https://www2.ed.gov/admins/finaid/accred/accreditation.html#Overview

Watkins, C. L. (1997). *Project Follow Through: A case study of contingencies influencing instructional practices of the educational establishment.* Cambridge Center for Behavioral Studies.

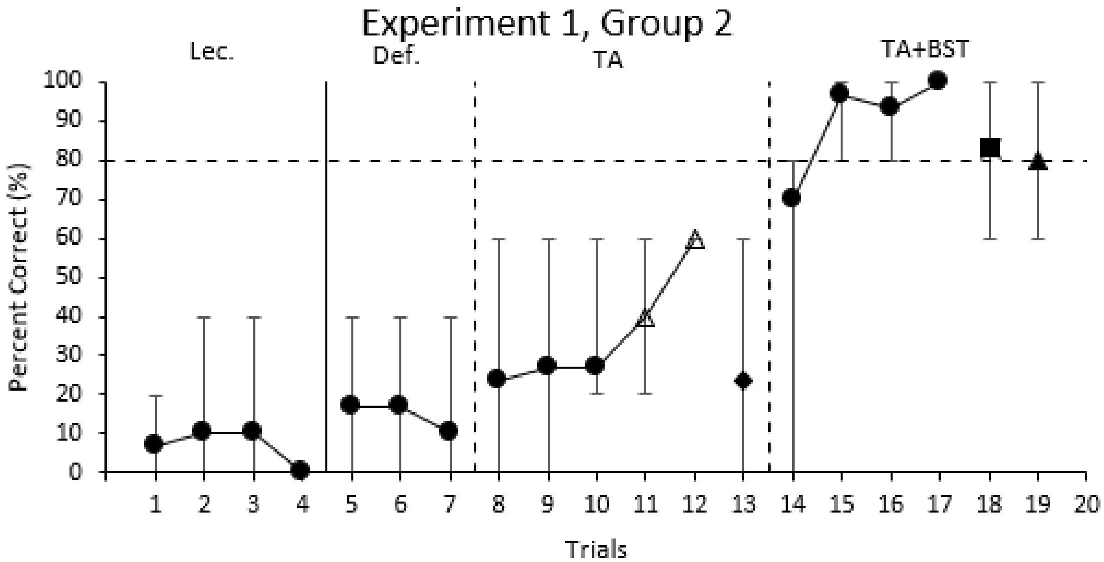
Whiting, S. W., Miller, J. M., Hensel, A. M., Dixon, M. R., & Szekely, S. (2014). Increasing the accuracy of EpiPen administration with a brief behavioral skills training package in a school for autism. *Journal of Organizational Behavior Management*, *34*(4), 265-278, https://doi.org/10.1080/01608061.2014.973632

Zayac, R. M., Ratkos, T., Frieder, J. E., & Paulk, A. (2016). A comparison of active student responding modalities in a general psychology course. *Teaching of Psychology*, *43*(1), 43–47. https://doi.org/10.1177/0098628315620879



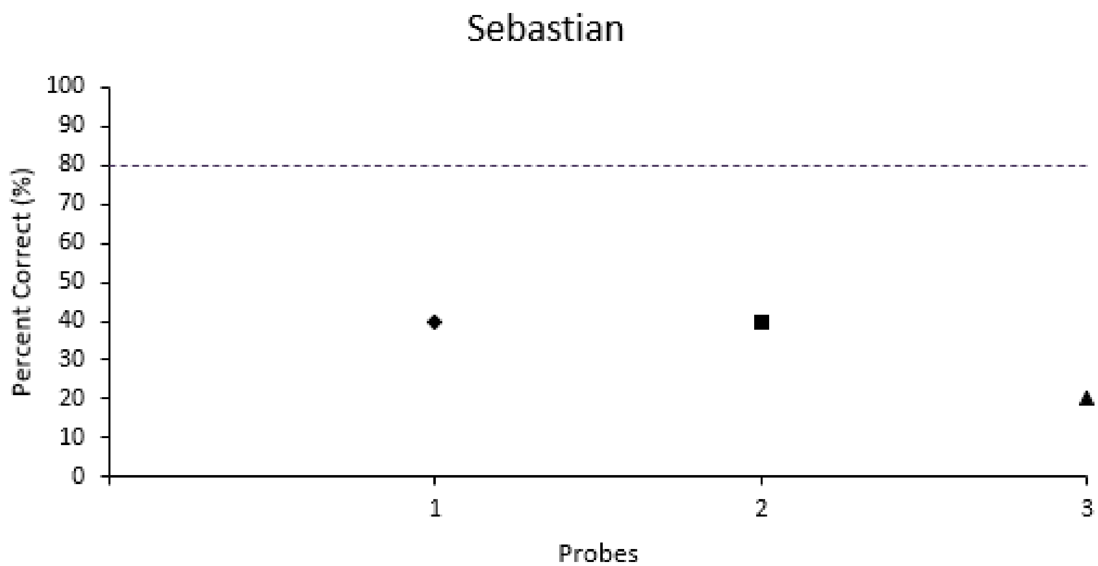
Note: Hollow triangles indicate averaged data across less than half of the total participants
Solid diamond, square, and triangle indicate probes

Figure 2



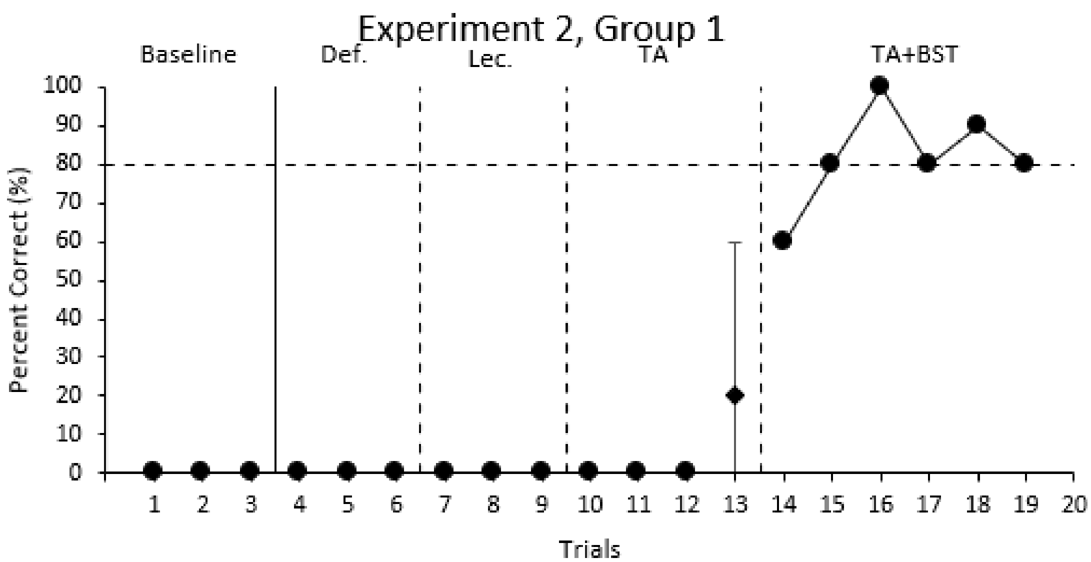
Note: Hollow triangles indicate averaged data across less than half of the total participants
Solid diamond, square, and triangle indicate probes

Figure 3



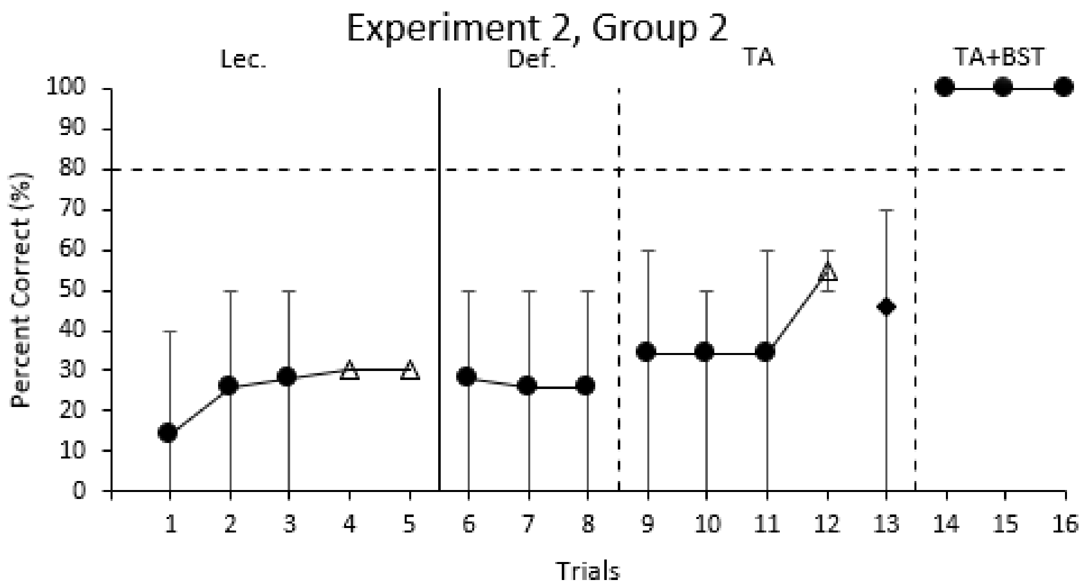
Note: Probe data for Experiment 1, Group 3 (Control)

Figure 4



Note: A solid diamond indicates a probe

Figure 5



Note: Hollow triangles indicate averaged data across less than half of the total participants
A solid diamond indicates a probe

Table 1

Demographic & Background Data For Participants of Experiment 1

Group 1	Age	Gender	Ethnicity	Education Level
Alistair	18	Man	White	High School
Sten	22	Man	BIPoC	High School
Oghren	18	Man	BIPoC	High School
Morrigan	20	Woman	White	High School
Zevran	18	Man	BIPoC	High School
Anders	18	Man	BIPoC	High School
Leliana	19	Woman	White	High School
Nathaniel	20	Man	BIPoC	High School
Sigrun	20	Man	BIPoC	High School
Orson	18	Man	White	High School
Silas	17	Man	BIPoC	High School
Wynne	30	Woman	White	Associates
Brogan	18	Man	White	High School
Dace	18	Man	White	High School
Group 2	Age	Gender	Ethicity	Education Level
Mhairi	18	Woman	White	Associates
Jerrik	16	Man	BIPoC	High School
Finn	19	Man	BIPoC	High School
Hawke	18	Man	White	High School
Carver	21	Man	BIPoC	Bachelors
Sebastian	19	Man	White	High School
Group 3	Age	Gender	Ethicity	Education Level
Fenris	18	Man	White	High School

Note: BIPoC = Black, Indigenous, & People of Color

Table 2

Intake Survey Demographic Data for Experiment 1

		HS	AA	BA	MA	PhD	Oth.	
1.	What is the highest level of education that you completed?	18	2	1	-	-	-	
		CS	SE	Mth	Eng.	Oth.	N/A	
2.	What is your major?	9	4	2	3	3	-	
3.	What is your minor?	4	-	-	-	1	16	
		Yes	No	IDK				
4.	Have you taken a college-level class before this semester? (During High school or previous semester)	15	6	-				
5.	Have you ever written a line of code?	21	-	-				
6.	Have you ever used a command line?	21	-	-				
7.	Are you fluent in any programming language?	6	10	5				
8.	Have you ever built your own computer?	5	16	-				
		N/A	1M	3M	6M	1Y	2Y	3Y+
9.	How much programming experience do you have?	1	4	2	3	6	3	2
10.	How much web development/design experience do you have?	10	4	5	2	-	-	-
11.	How many years of general work experience do you have? (Any Job)	8	-	2	2	2	2	5
12.	Which of the following classes have you taken or are taking?	Alg	Geo	PreC	Dis	Stat	Phy1	PM
		17	17	14	15	11	12	7
		Alg2	Trig	Cal	Cal2	Cal3	Phy2	Oth.
		16	14	19	10	5	6	3
		L.A.	Dif.	Ch	Ch2			N/A
		6	3	15	6			1

Note: HS = High School, AA = Associate's, BA = Bachelor's, MA = Master's, PhD = Doctorate, Oth. = Other, CS = Computer Science, SE = Software Engineering, Mth = Math, Eng. = Engineering, ME = Mechanical Engineering, IDK = I don't know, M = Month, Y = Year, Alg = Algebra, Geo = Geometry, PreC = Pre-Calculus, Dis = Discrete Mathematics, Stat = Statistics & Probability, Phy = Physics, PM = Physics & Mechanics, Trig = Trigonometry, Cal = Calculus, L.A. = Linear Algebra, Dif. = Partial Differential Equasions, Ch = Chemistry.

Table 3

Average IOA Across Participants & Phases for Experiment 1, Group 1.

	BL	Def.	Lec.	TA	TA+ BST	AS 2	AS 3	AS 4	Avg. Agmt.
Brogan	100	100	100	93.33	100	100	100	100	99.17
Morrigan	95.00	93.33	100	100	100	100	100	100	98.54
Orson	100	100	100	100	100	100	100	100	100
Wynne	100	100	100	100	100	100	-	-	100
Nathaniel	100	100	100	100	100	100	-	-	100
Silas	100	100	100	93.33	100	100	100	100	99.17
Alistair	100	100	100	100	100	100	100	100	100
Zevran	100	100	100	100	96.00	100	100	100	99.50
Oghren	100	100	100	100	96.00	100	-	-	99.33
Lelianna	100	100	100	93.33	95.00	100	100	100	98.54
Sigrun	100	100	93.33	93.33	100	100	100	100	98.33
Anders	100	100	100	93.33	100	100	100	100	99.17
Dace	100	100	100	100	100	100	100	100	100
Sten	100	100	100	92.00	95.00	100	100	100	98.38
Avg. Agmt.	99.64	99.52	99.52	97.05	98.71	100	100	100	99.30

Note: BL = Baseline, Def. = Technical Definition, Lec. = In-Class Lecture, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe

Table 4

Average IOA Across Participants & Phases for Experiment 1, Group 2.

	BL	Def.	TA	TA+ BST	AS 2	AS 3	AS 4	Avg. Agmt.
Jerrik	100	100	100	100	80.00	100	100	97.14
Hawke	100	100	96.00	100	100	100	100	99.43
Finn	100	100	100	100	100	100	100	100
Mhairi	100	100	100	95.00	80.00	100	100	96.43
Sebastian	93.33	100	100	100	100	80.00	80.00	93.33
Carver	100	100	100	95.00	100	100	100	99.29
Avg. Agmt.	98.89	100	99.33	98.33	93.33	96.67	96.67	97.60

Note: BL = Baseline, Def. = Technical Definition, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe

Table 5

Score Distribution Across Number and Percentage of Participants for Experiment 1, Group 1.

Score	0		20		40		60		80		100		Total Participants
Participants	#	%	#	%	#	%	#	%	#	%	#	%	
BL	13	93	1	7	-	-	-	-	-	-	-	-	14
Def.	12	86	-	-	2	14	-	-	-	-	-	-	
Lec.	10	71	1	7	2	14	1	7	-	-	-	-	
TA	6	43	-	-	2	14	3	21	3	21	-	-	
AS2	6	55	1	9	2	18	3	27	1	9	1	9	
TA+BST	-	-	-	-	-	-	-	-	2	14	12	86	
AS3	1	9	-	-	-	-	2	18	1	9	7	64	10
AS4	1	9	-	-	-	-	1	9	1	9	8	73	

Note: BL = Baseline, Def. = Technical Definition, Lec. = In-Class Lecture, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe

Table 6

Score Distribution Across Number and Percentage of Participants for Experiment 1, Group 2.

Score	0		20		40		60		80		100	
Participants	#	%	#	%	#	%	#	%	#	%	#	%
BL	4	67	1	17	1	17	-	-	-	-	-	-
Def.	3	50	1	17	2	33	-	-	-	-	-	-
TA	2	33	1	17	2	33	1	17	-	-	-	-
AS2	3	50	1	17	-	-	2	33	-	-	-	-
TA+ BST	-	-	-	-	-	-	-	-	1	17	5	83
AS3	-	-	-	-	-	-	2	33	1	17	3	50
AS4	-	-	-	-	-	-	3	50	0	0	3	50

Note: BL = Baseline, Def. = Technical Definition, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe
N = 6 in experiment 1, group 2. The score distribution displayed on this table is based on the average across a stable range of data for each phase.

Table 7

Social Validity Survey Data for Experiment 1

		SD	D	N	A	SA
1.	I enjoyed participating in the study.	2	-	1	4	4
2.	The description of the study was accurate.	1	1	2	2	5
3.	Participating in the study was worth 5% extra credit.	2	-	-	2	7
4.	The time spent participating in the study was worthwhile.	2	-	1	3	5
5.	I was comfortable with where the study sessions were conducted.	2	-	1	2	6
6.	The instructions I received during the study were clear.	2	-	1	2	6
7.	The feedback I received on my algorithms was fair and objective.	2	-	1	3	5
8.	The study improved my ability to write programs.	2	-	2	3	4
9.	The study improved my ability to write algorithms.	2	-	2	-	7
10.	I better understand the distinction between programs and algorithms because of the study.	2	-	-	2	7
11.	The instructions I received during the study made sense.	1	1	1	2	6
12.	My grade in the class improved because of my participation in the study.	1	1	3	4	2
13.	I am a better computer scientist because of the study.	1	1	1	3	5
14.	Algorithms are important.	2	-	-	3	6
15.	The ability to write algorithms is valuable.	2	-	-	3	6
16.	The feedback I received on my algorithms was beneficial.	2	-	1	4	4
17.	Practice helped me improve my algorithm writing ability.	2	-	2	2	5
18.	Having a rubric/task analysis when I write algorithms is useful.	2	-	1	4	4
19.	Having the teacher/researcher demonstrate writing an algorithm was valuable.	2	-	1	3	5
20.	Instruction I received in the study differed significantly from instruction I received in class.	2	2	-	6	1

Note: SD = Strongly Disagree, D = Disagree, N = Neutral, A = Agree, SA = Strongly Agree

Table 8

Demographic & Background Data For Participants of Experiment 2

Group 1	Age	Gender	Ethnicity	Education Level
Solas	24	Man	BIPoC	High School
Sera	19	Man	White	High School
Blackwall	18	Man	White	High School
Group 2	Age	Gender	Ethnicity	Education Level
Gorim	19	Man	White	Associates
Daveth	20	Man	White	High School
Renn	19	Man	BIPoC	High School
Velanna	19	Woman	BIPoC	High School
Gilmore	20	Man	White	High School

Note: BIPoC = Black, Indigenous, & People of Color

Table 9

Intake Survey Demographic Data for Experiment 2

		HS	AA	BA	MA	PhD	Oth.	
1.	What is the highest level of education that you completed?	7	1	-	-	-	-	
		CS	SE	Mth	Eng.	Oth.	N/A	
2.	What is your major?	2	1	1	3	1	-	
3.	What is your minor?	2	-	-	-	1	5	
		Yes	No	IDK				
4.	Have you taken a college-level class before this semester? (During High school or previous semester)	7	1	-				
5.	Have you ever written a line of code?	8	-	-				
6.	Have you ever used a command line?	7	-	1				
7.	Are you fluent in any programming language?	2	5	-				
8.	Have you ever built your own computer?	4	4	-				
		N/A	1M	3M	6M	1Y	2Y	3Y+
9.	How much programming experience do you have?	1	4	1	-	2	-	-
10.	How much web development/design experience do you have?	6	-	2	-	-	-	-
11.	How many years of general work experience do you have? (Any Job)	3	-	-	1	1	1	2
12.	Which of the following classes have you taken or are taking?	Alg	Geo	PreC	Dis	Stat	Phy1	PM
		4	4	7	3	3	2	3
		Alg2	Trig	Cal	Cal2	Cal3	Phy2	Oth.
		4	4	8	2	1	2	-
		L.A.	Dif.	Ch	Ch2			N/A
		1	1	4	-			-

Note: HS = High School, AA = Associate's, BA = Bachelor's, MA = Master's, PhD = Doctorate, Oth. = Other, CS = Computer Science, SE = Software Engineering, Mth = Math, Eng. = Engineering, ME = Mechanical Engineering, IDK = I don't know, M = Month, Y = Year, Alg = Algebra, Geo = Geometry, PreC = Pre-Calculus, Dis = Discrete Mathematics, Stat = Statistics & Probability, Phy = Physics, PM = Physics & Mechanics, Trig = Trigonometry, Cal = Calculus, L.A. = Linear Algebra, Dif. = Partial Differential Equasions, Ch = Chemistry.

Table 10

Average IOA & TI Across Participants & Phases for Experiment 2, Group 1.

	BL		Def.		Lec.		TA		TA+ BST		AS 2		Avg. Agmt.	
	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI
Blackwall	100	100	100	100	100	100	100	100	-	-	100	100	100	100
Sera	100	100	100	100	100	100	100	100	90.00	100	100	100	98.33	100
Solas	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Avg. Agmt.	100	100	100	100	100	100	100	100	95.00	100	100	100	99.26	100

Note: BL = Baseline, Def. = Technical Definition, Lec. = In-Class Lecture, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe

Table 11

Average IOA & TI & Percent IOA & TI Collected Across Participants & Phases for Experiment 2, Group 2.

	BL		Def.		TA		TA+ BST		AS 2		Avg. Agmt.		% of	
	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI	IOA	TI
Gilmore	100	100	100	100	100	100	-	-	100	100	100	100	100	80.00
Daveth	100	100	96.67	100	100	100	100	100	90.00	100	97.33	100	100	100
Gorim	92.00	100	100	100	92.50	100	-	-	100	100	96.13	100	100	100
Renn	100	-	90.00	100	95.00	100	-	-	100	100	96.25	100	100	72.73
Velanna	96.67	100	100	100	100	100	-	-	100	100	99.17	100	100	100
Avg. Agmt.	97.73	100	97.33	100	97.50	100	100	100	98.00	100	97.94	100	-	-
% of	100	82.35	100	86.67	100	100	100	100	100	100	-	-	100	92.17

Note: BL = Baseline, Def. = Technical Definition, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe

Table 12

Score Distribution Across Number and Percentage of Participants for Experiment 2, Group 1.

Score	0		10		20		30		40		50		60		70		80		90		100		Total Participants
Participants	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	
BL	3	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Def.	3	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Lec.	3	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
TA	3	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
AS2	2	67	-	-	-	-	-	-	-	-	-	-	1	33	-	-	-	-	-	-	-	-	
TA+ BST	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	100	-	-	2

Note: BL = Baseline, Def. = Technical Definition, Lec. = In-Class Lecture, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe
The score distribution displayed on this table is based on the average across a stable range of data for each phase.

Table 13

Score Distribution Across Number and Percentage of Participants for Experiment 2, Group 2.

Score	0		10		20		30		40		50		60		70		80		90		100		Total Participants
Participants	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	#	%	
BL	2	40	-	-	-	-	1	20	1	20	1	20	-	-	-	-	-	-	-	-	-	-	
Def.	1	20	1	20	-	-	1	20	1	20	1	20	-	-	-	-	-	-	-	-	-	-	5
TA	1	20	-	-	1	20	-	-	1	20	1	20	1	20	-	-	-	-	-	-	-	-	
AS2	1	20	-	-	-	-	-	-	1	20	-	-	2	40	1	20	-	-	-	-	-	-	
TA+ BST	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	100	1

Note: BL = Baseline, Def. = Technical Definition, TA = Task Analysis, BST = Behavioral Skills Training, AS = Assignment Probe
The score distribution displayed on this table is based on the average across a stable range of data for each phase.

Table 14

Social Validity Survey Data for Experiment 2

		SD	D	N	A	SA
1.	I enjoyed participating in the study.	-	-	-	2	5
2.	The description of the study was accurate.	-	-	1	1	5
3.	Participating in the study was worth 5% extra credit.	-	-	-	2	5
4.	The time spent participating in the study was worthwhile.	-	-	-	1	6
5.	I was comfortable with where the study sessions were conducted.	-	-	-	1	6
6.	The instructions I received during the study were clear.	-	-	1	1	5
7.	The feedback I received on my algorithms was fair and objective.	-	-	2	1	4
8.	The study improved my ability to write programs.	1	-	-	1	5
9.	The study improved my ability to write algorithms.	-	-	-	2	5
10.	I better understand the distinction between programs and algorithms because of the study.	-	-	1	2	4
11.	The instructions I received during the study made sense.	-	-	2	3	2
12.	My grade in the class improved because of my participation in the study.	1	-	1	1	4
13.	I am a better computer scientist because of the study.	1	-	-	1	5
14.	Algorithms are important.	-	2	-	-	5
15.	The ability to write algorithms is valuable.	-	-	1	-	6
16.	The feedback I received on my algorithms was beneficial.	-	-	1	2	4
17.	Practice helped me improve my algorithm writing ability.	-	-	1	2	4
18.	Having a rubric/task analysis when I write algorithms is useful.	-	-	1	2	4
19.	Having the teacher/researcher demonstrate writing an algorithm was valuable.	-	-	-	2	5
20.	Instruction I received in the study differed significantly from instruction I received in class.	2	1	-	3	1

Note: SD = Strongly Disagree, D = Disagree, N = Neutral, A = Agree, SA = Strongly Agree

Appendix A

Experimental 1 Task Analysis

1. Read the problem/program specifications

2. Identify the key given information. (Underline, highlight, list, etc.)

3. Identify specified input and output.

4. Identify some components it must have

5. Identify something similar or related to the problem

6. Organize algorithm systematically (numbered steps/visuals or flow charts

Appendix B

Data Collect and Work Problem Example

---Researcher Section---

1: A.___ B.___ C.___ Participant Initials: ____
2: A.___ B.___ C.___ Researcher Initials: ____
3: A.___ B.___ C.___ Date:_________
4: A.___ B.___ C.___ (P1)(P2)(P3) : (BL1)(BL2)(BL3) (Tx1)(Tx2) (A)
5: A.___ B.___ C.___ / T :___ / G: (Y) (N) Series:_________

---Student Section---

Circle one - The researcher presented a task aid: (N/A) (A) (B) (C) (D) (E) (Feedback)

1. Write an algorithm for a program that prints out “Hello, World.” to the screen.

Appendix C

Experiment 2 Task Analysis

Task Aid: (D)

- ➢ **Key information:**
  - ✓ Explicitly list or highlights key information from task prompt (List, highlight, underline, bold, italicize, etc.)
  - ✓ Discusses key information in enough detail for others to follow
- ➢ **Input/Output:**
  - ✓ Input: What is the input? Is there an input? Who, what, when, where, why, how?
  - ✓ Output: What is the output? Is there an output? Who, what, when, where, why, how?
- ➢ **Organization:**
  - ✓ Sequential/orderly steps (Start to finish)
  - ✓ Structure (Numbered steps, diagrams, visuals, flow charts, etc)
- ➢ **What must be used to accomplish each step:**
  - ✓ Base level: Variable declaration, libraries, main parameters (int main void, return 0),
  - ✓ Higher-level: Standard functions (printf/scanf), boolean (if/else), loops (while, do-while, for), data structures (arrays/structs), pointers.
- ➢ **Sub-problem:**
  - ✓ Components are broken down into individual pieces
  - ✓ References/visualizes the desired result or something similar (Desired output, diagram, code trace, pseudo code)

Appendix D

Experimental Design Proposal

Based on the study above, participants researchers would recruit from the same CSE1001 Fundamentals of Software Development 1 course given similar conditions. Researchers would recruit three to five participants without an incentive component for participation and screen them for prior experience with writing algorithms. The locations, data collection, work problems, dependent variable, and mastery criterion would be the same as Experiment 2. The only difference regarding sessions is that sessions would co-occur for all participants weekly. Additionally, the researchers would control the implementation of independent variables. The experimental design would be a component analysis of behavioral skills training using concurrent multiple-baseline across participants. There would be four experimental phases, including; instruction, modeling, rehearsal, and feedback.

All participants would start at baseline at the same time completing work problem trials. Once a participant completes at least three trials that indicate consistent performance, researchers will implement instruction. At the same time, the other two participants will continue in the baseline. Once data for a second participant is consistent for at least three more trials since the first participant entered the instruction phase, the second participant will enter the instruction phase. The final participant will remain at baseline and enter into the instruction phase after data continue to show consistency for at least three more trials since the second participant entered instruction.

Each participant will continue this cycle for each phase through instruction, modeling, rehearsal, and feedback. Researchers will ensure that there is adequate stability and overlap within the data before making any experimental changes. Once all the experimental phases are complete, researchers will give participants a debrief and a social validity survey.