# USING SUBGOAL LEARNING AND SELF-EXPLANATION TO IMPROVE PROGRAMMING EDUCATION

A Dissertation
Presented to
The Academic Faculty

By

Lauren E. Margulieux

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
School of Psychology

Georgia Institute of Technology

May 2016

Using Subgoal Learning and Self-Explanation to Improve Programming Education

Approved by:

Dr. Richard Catrambone, Advisor
School of Psychology
*Georgia Institute of Technology*

Dr. Francis T. Durso
School of Psychology
*Georgia Institute of Technology*

Dr. Mark Guzdial
School of Interactive Computing
*Georgia Institute of Technology*

Dr. Wendy Newstetter
College of Engineering
*Georgia Institute of Technology*

Dr. Wendy A. Rogers
School of Psychology
*Georgia Institute of Technology*

Date Approved: March 18, 2016

**ACKNOWLEDGEMENTS**

I would like to thank my committee members and my family for always pushing me to be better.

**TABLE OF CONTENTS**

# LIST OF TABLES

## LIST OF FIGURES

# SUMMARY

The present study combined subgoal learning and self-explanation frameworks to improve problem solving performance. Subgoal learning has been used to promote retention and transfer in procedural domains, such as programming. The primary method for learning subgoals, however, has been through passive learning methods, and passive learning methods are typically less effective than constructive learning methods. To promote constructive methods of learning subgoals, a subgoal learning framework was used to guide self-explanation. Self-explanation is an effective method for engaging learners to make sense of new information based on prior knowledge and logical reasoning. Self-explanation is typically more effective when learners receive some guidance, especially if they are novices, because it helps them to focus their attention on relevant information. In the present study, only some of the constructive learning methods produced better problem solving performance than passive learning methods. Learners performed best when they learned constructively and either received hints about the subgoals of the procedure or received feedback on the self-explanations that they constructed, but not when they received both hints and feedback. When students received both types of guidance, they did not perform better than those who learned subgoals through passive learning methods. These findings suggest that constructive learning of subgoals can further improve the benefits of learning subgoals, but there is an optimal level of guidance for students engaging in constructive learning. Providing too much guidance can be as detrimental as providing too little. This nuance is important for educators who engage their students in constructive learning and self-explanation to recognize and promote the best results.

# CHAPTER 1: INTRODUCTION

Students in higher education need to be able to learn independently, at least in part. As the number of students pursuing bachelors and advanced degrees increases, so does the ratio of students to instructors and the number of online courses. These factors make direct interaction between students and instructors increasingly limited and self-guided learning increasingly valuable. To help students be more independent learners, support from researchers and instructional designers is needed. The present research examined a new strategy to support independent learning: the integration of subgoal learning and self-explanation.

## 1.1 Subgoal Learning

Subgoal learning refers to a strategy used predominantly in STEM fields that helps students to deconstruct problem solving procedures into subgoals. Deconstructing procedures into subgoals helps learners to better recognize the structural components of the problem solving process (Atkinson, Catrambone, & Merrill, 2003; Catrambone, 1998). Subgoals are functional pieces of procedures used to solve problems that contain one or more individual steps. They are inherent in all procedures except the most basic. For instance, if algebra students were asked to solve the equation in Figure 1 for *x*, they would likely start by isolating terms with *x*s on one side of the equation and the others on the opposite side. Then they would simplify the terms until *x* had a coefficient of one. Isolating and simplifying terms are subgoals of the procedure used to solve for a variable. They are functional parts, each containing two independent steps, that are necessary to solve this class of problems.

*Solve for x*

$$4x - 8 = 2x + 6$$

$$\left.\begin{array}{ll} +8 & +8 \\ -2x & -2x \end{array}\right\} \text{Subgoal: Isolate variable}$$

$$4x - 2x = 6 + 8$$

$$\left.\begin{array}{l} 2x = 14 \\ /2 \quad /2 \end{array}\right\} \text{Subgoal: Simplify terms}$$

$$x = 7$$

*Figure 1.* Worked example of the procedure used to solve for a variable. Steps of the worked example are grouped into the subgoals, denoted by brackets, necessary for solving problems in this class.

Research suggests that when instructions help students learn the subgoals of a procedure, students are better able to transfer knowledge to solve novel problems. In some of the original research on subgoal learning, Catrambone and Holyoak (1990) found that when instructional materials highlighted the subgoals of a procedure, learners were more likely to correctly apply them to problems that used the same procedure but had different contextual features (e.g., problems about apples versus those about oranges) or had modified or new steps. Subsequent studies (Atkinson, 2001; Catrambone, 1994, 1995, 1996, 1998; Margulieux & Catrambone, 2014; Margulieux, Guzdial, & Catrambone, 2012) have consistently found that subgoal-oriented instructions improved problem solving performance across a variety of STEM domains, such as programming (e.g., Margulieux et al., 2012) and statistics (e.g., Catrambone, 1998).

Subgoal-oriented instructions are typically implemented as worked examples. Worked examples give learners concrete examples of the procedure being used to solve a problem. Because problems necessarily include a context, such as apples or oranges, worked examples include context-specific information. The other main type of instruction used to teach procedures is expository instruction, which gives context-independent information about the problem solving procedure (LeFevre & Dixon, 1986). Both types of instructions are important for efficient learning, but learners tend to focus more attention on worked examples than expository instruction (Eiriksdottir & Catrambone, 2011). LeFevre and Dixon (1986) found that when expository text and worked examples described different procedures, 92% of participants used the procedure demonstrated through worked examples rather than that described in the expository text. This finding and the fact that few participants in LeFevre and Dixon's study asked questions about the conflicting procedures suggests that the learners favored the worked examples over expository instruction.

Eiriksdottir and Catrambone (2011) argued that learning primarily from worked examples does not inherently promote deep processing of concepts. While it may result in better initial performance because examples are more easily mapped to similar problems, it is less likely to result in retention and transfer (Eiriksdottir & Catrambone, 2011). When studying examples, learners tend to focus on superficial features rather than the structural features because superficial features are easier to grasp and novices do not have the necessary domain knowledge to recognize the structural features of examples (Chi, Bassok, Lewis, Reimann, & Glaser, 1989). For instance, when studying physics worked examples, learners were more likely to remember that the example has a ramp than that the example used Newton's second law (Chi et al., 1989). Novices' focus on superficial features leads to ineffective organization and storage of information that, in turn, leads to ineffective recall and transfer (Bransford, Brown, & Cocking, 2000).

To promote deeper processing of worked examples and, thus, improve retention and transfer, worked examples have been manipulated to promote subgoal learning. In particular, subgoal labeling is a technique used to promote subgoal learning and to help learners recognize the structure of procedures exemplified in worked examples (e.g., Catrambone, 1994, 1995, 1996, 1998). Subgoal labels are function-based instructional explanations that describe the purpose of a subgoal to the learner. For instance, for the problem in Figure 1 and for the subgoal in which the problem solver isolates terms with *x*s one on side the equation, the subgoal label might read "Isolate variable." This label provides information about the purpose of that subgoal and the function behind the steps within it.

Studies have found that receiving subgoal labels in worked examples improves performance while solving novel problems, and it does not increase the amount of time learners spend studying instructions or solving problems (e.g., Margulieux et al., 2012). Subgoal labels are believed to be effective because they visually group the steps of worked examples into subgoals and meaningfully label those groups (Atkinson et al., 2003). This format highlights the structure of examples, helping students focus on structural features and more effectively organize information (Atkinson, Derry, Renkl, & Wortham, 2000; Catrambone, 1995, 1996, 1998). Catrambone (1995) found that when participants were asked to explain solutions to problems after receiving either subgoal labeled or unlabeled examples, those who received subgoal labels grouped solutions into subgoals and used the subgoal labels in their explanations. This finding suggests that subgoal labels prompted students to organize information around the subgoals of the procedure and that students adopted the labels to articulate the procedure.

By helping learners to organize information and to focus on structural features of worked examples, subgoal labels are believed to reduce the extraneous cognitive load that is inherent in worked examples and can hinder learning (Renkl & Atkinson, 2002). Extraneous cognitive load is the cognitive load associated with information in instructions that is not necessary to learning the procedure (Sweller, 2010). Worked examples introduce extraneous cognitive load because they contain information specific to a context, and students must process this superficial information about the context even though it is not relevant to the underlying procedure (Sweller, 2010). Subgoal labels can reduce focus on superficial features by highlighting the structural features of the procedure (Renkl & Atkinson, 2002). Reducing extraneous cognitive load allows more mental resources to be devoted to learning the procedure through building schemas, chunking information, and connecting prior knowledge and new knowledge (Sweller, 2010). Subgoal labels further support these processes by providing a mental organization (i.e., subgoals) for storing information.

Subgoal labels that are independent from a specific context have been the most effective type of subgoal labels in the past (Catrambone, 1995, 1998). Catrambone (1998) found that learners who had prior knowledge in the domain performed better on problem solving tasks that were given after a week-long delay or that required using the procedure differently than demonstrated in the examples when they received labels that were abstract (e.g., **Ω**) compared to when they received labels that were context-specific (e.g., isolate *x*). Catrambone (1998) argued that learners with sufficient prior knowledge were able to correctly explain to themselves the purpose of the subgoal. He argued, therefore, that prompting self-explanation of the subgoal's function by providing a label that did not explain the subgoal's function was more effective than providing an informative label.

The findings from Catrambone's (1998) study align with a growing body of evidence that learning is more effective when students actively or constructively engage with content rather than passively receive content. This body of evidence is summarized by Chi (2009) and used to support her Interactive-Constructive-Active-Passive (ICAP) framework. In this framework, Chi (2009) characterized four types of learning based on students' engagement with content: interactive, constructive, active, and passive (see Figure 2 for definitions and examples).

| | **Passive** | **Active** | **Constructive** | **Interactive** |
|---|---|---|---|---|
| **Definition** | Receiving information without physical activity | Receiving information with physical activity | Individually producing information beyond that which is provided | Collaboratively producing information beyond that which is provided |
| **Examples** | Listen to a lecture<br>Read a textbook | Taking notes on a lecture<br>Highlighting sections of a reading | Connecting concepts to prior knowledge<br>Explaining the steps of a worked example | Discussing a concept<br>Providing and responding to peer feedback |

*Figure 2.* Definitions and characteristics of passive, active, constructive, and interactive learning based on the ICAP framework proposed by Chi (2009).

Using this framework to compare the learning outcomes from various learning activities, Chi (2009) found that interactive and constructive learning were the most effective, active learning was the second most effective, and passive learning was the least effective. Most research about subgoal learning, besides Catrambone (1998), has provided meaningful subgoal labels that explain the function of subgoals to learners, making this learning passive. The present study explored whether more engaging methods of learning subgoals would improve novel problem solving.

### 1.2 Self-Explanation

Self-explanation is a common and effective type of active or constructive learning that might help students to learn subgoals (Chi, 2009). Self-explanation is a learning strategy in which students use prior knowledge and logical reasoning to make sense of new information and gain new knowledge. A recent review of self-explanation studies found that this strategy is effective across a range of domains, as long as the domain has logical rules with few exceptions (Wylie & Chi, 2014). For example, self-explanation has been highly effective in physics education, which has rule-based, procedural problem solving, but it has not been effective in language education, which has several exceptions to grammatical and orthoepic rules.

Self-explanation of worked examples in procedural domains, which typically follow logical rules, generally improves learning outcomes (Wylie & Chi, 2014). Similar to subgoal learning, self-explanation of worked examples identifies which features are structural and reasons about the function of steps (Bielaczyc, Pirolli, & Brown, 1995; Chi, de Leeuw, Chiu, & LaVancher, 1994). Moreover, self-explanation is believed to be effective for many of the same reasons as subgoal learning. By self-explaining worked examples, learners recognize which features are structural and which are superficial to the example. By recognizing which features are most important, learners can reduce mental resources devoted to processing superficial features and reduce extraneous cognitive load, allowing for more learning processes (Renkl & Atkinson, 2002). Self-explanation further improves learning processes because it helps students to activate relevant prior knowledge and integrate new information with prior knowledge (Chi et al., 1994; Sweller, 2010). These processes help learners build better a mental representation of the procedure that allows them to more easily apply their knowledge to novel problems (Renkl & Atkinson, 2003).

Self-explanation can be an active or constructive type of learning. Active self-explanation typically involves learners selecting explanations from a list of possible explanations (e.g., Aleven & Koedinger, 2000; Conati & VanLehn, 2000). This method requires activity from the learner but not construction, which matches Chi's (2009) definition of active learning. It is important to note that this definition is different than the more general definition of active learning that includes methods that Chi would consider active, constructive, and interactive. Chi's definitions were used for the present study because they distinguish between learning methods that require activity and those that require construction. Constructive self-explanation requires learners to create explanations for themselves (e.g., Chi et al., 1989; Bielaczyc et al., 1995; Schworm & Renkl, 2006).

Both active and constructive self-explanation offer integration and cognitive load benefits, but constructive explanation is considered to have additional benefits because it requires learners to generate an explanation. The generation effect states that learners remember information better when they produce it rather receive it (Jacoby, 1978). As deWinstanly, Bjork, and Bjork (1996) argued, the generation effect works because the cognitive processes involved in encoding information are similar to those involved in retrieving information; therefore, the learner has the same cues while retrieving information as they had while encoding it. In their review of self-explanation literature, Wylie and Chi (2014) found that constructive self-explanation was more effective than active self-explanation, if learners successfully engage in it.

Learners do not always, or even commonly, successfully engage in constructive self-explanation on their own (Chi et al., 1989; Renkl, 2005). Chi et al. (1989) found that about 10% of learners self-explained examples without external prompting. Many studies have replicated this low percentage or found even fewer learners are truly constructively self-explaining instead of engaging in a shallower level of processing, such as paraphrasing instructions (e.g., Hausmann & Chi, 2002). Renkl and colleagues (1998, 2005) argued that many learners do not self-explain, especially when they have little prior knowledge, because it requires a large amount of effort and mental resources. Learners who self-explain must process new information, recall prior knowledge, and reason about which pieces of information are relevant and critical for the procedure. Learners can self-explain if they devote additional time to the task (Wylie & Chi, 2014), but they need to be reminded to do so. To remind learners to self-explain, many methods of prompting self-explanation have been developed.

Research has found little difference in the learning outcomes of students who are intrinsically or extrinsically motivated to self-explain, suggesting that self-explanation itself leads to learning benefits rather than characteristics of students who self-explain (e.g., Bielaczyc et al., 1995; Chi et al., 1994; Hausmann & Chi, 2002; Renkl, Stark, Gruber, & Mandl, 1998). Bielaczyc et al. (1995) found that learners who were trained to self-explain had equivalent learning outcomes as those who self-explained without training. Renkl et al. (1998) found similar benefits of self-explanation training but also that training had only a short-term effect on participants' number of self-explanations and performance. For lasting effects, Renkl et al. (1998) found that instructions needed to include prompts to self-explain, because prompts gave reminders throughout the instructions. In fact, prompted self-explanation sometimes leads to better learning than unprompted self-explanation because the prompts focus learners' attention to important information and reduce inaccurate explanations (Chi et al., 1994; Renkl, 1997, 2002).

Self-explanation prompts ask learners to complete tasks that range from unstructured to structured. Unstructured prompts range from answering completely open-ended questions, like "Can you explain that?" (Hausmann & Chi, 2002), to answering focused questions, like "Explain how examples 1 and 2 are similar," (de Koning, Tabbers, Rikers, & Paas, 2011). Structured prompts range from filling in blanks of partial explanations (Berthold, Eysink, & Renkl, 2009), to selecting explanations from a menu (Conati & VanLehn, 2000). Most of the prompts in this spectrum require constructive learning because students construct their own explanations, but prompts that ask students to select an explanation from a menu require only active learning. Though constructive learning is generally more effective than active learning (Chi, 2009), the most effective type of self-explanation prompt depends on characteristics of the learner.

To develop the best type of self-explanation prompt, the amount of information in the prompt needs to be balanced with the learners' prior knowledge (Renkl, 2002). In their review of the self-explanation literature, Wylie and Chi (2014) found that self-explanation is not effective if learners do not have gaps in their understanding after instruction. In other words, if learners are given all of the knowledge that they need, nothing is left to construct. Therefore, prompts should not include so much information such that self-explanation is not necessary. If learners are given too little information, however, they spend too much of their cognitive capacity trying to figure out what they should be learning to actually learn (Kirschner, Sweller, & Clark, 2006). For example, Wylie and Chi (2014) found that focused self-explanation prompts were typically more effective than completely open-ended prompts. They argued that novices know so little about domains that they need clues about what to explain to be most effective. Focused prompts are specific to the instructions, but they do not need to be individualized to the learner. Hausmann and Chi (2002) found that learning outcomes were equivalent between learners who were given individualized prompts based on their prior self-explanations created in real-time by a human tutor and learners who were given pre-packaged prompts. This result suggests that, while self-explanation prompts should be tailored to the learner's general level of knowledge, they do not need to be tailored to each student to be effective.

### 1.3 Feedback

Similar to the amount of guidance learners receive while making explanations, the amount of feedback that learners should receive about their explanations depends on characteristics of the task and learners. In some situations, feedback on self-explanations can improve performance. For instance, Renkl (2002) found that learning outcomes were better when participants who self-explained worked examples could also access short instructional explanations to check their self-explanations than when participants could not access explanations. Renkl (2002) argued that the instructional explanations were necessary to reduce illusions of understanding and keep learners from perpetuating incorrect explanations. In other situations, however, feedback can hinder performance. For instance, Schworm and Renkl (2006) found that when learners were prompted to self-explain, learning outcomes were better without access to feedback than with access to feedback. They also found that learners who received feedback perceived their learning as greater, though they actually performed worse. Schworm and Renkl (2006) argued that learners overly relied upon the feedback and would not devote much effort to constructing self-explanations before seeking the explanations provided by the feedback. For these reasons, Schworm and Renkl (2006) argued that withholding feedback from learners might be more beneficial in some cases than ensuring that learners' self-explanations are correct.

One of the main differences between Renkl (2002) and Schworm and Renkl (2006) was the instructional domain. Renkl (2002) taught a statistical procedure for which learners generally made rationale-based self-explanations, and Schworm and Renkl (2006) taught instructional design for which learners generally made rule-based self-explanations. Rationale-based self-explanations describe why something was done, or the function that a step served. For instance, a learner explaining the worked example in Figure 1 might say that "8" was added to each side to isolate *x*. Rule-based self-explanations describe the rules of the domain that apply to a step. For Figure 1, a learner might explain that "8" was added to each side to follow the rule that states equal actions must be taken on both sides of the equation.

Rule- and rationale-based explanations have different uses in different types of domains. Rule-based self-explanations are more constructive in non-procedural domains, like design, than procedural domains because the rules can be applied in various ways (Berthold et al., 2009). For instance, the rule in design to group related information can be executed in different ways that results in different designs that all follow the rule. Therefore, making rule-based self-explanations helps the learner build a robust mental representation of various possible applications of the rules. The rules in procedural domains, however, are typically applied in one way. For instance, once the learner decides that “8” needs to be added to the left side of the equation in Figure 1, the algebraic rule to keep both sides of an equation balanced can be executed correctly in only one way, by adding “8” to the other side of the equation. Therefore, the rule simply needs to be memorized and implemented. To decide that “8” needs to be added to the left side of the equation, learners need to build a robust mental representation of the various applications of a procedure. In procedural domains, rationale-based explanations can construct this type of mental representation because they help the learner to recognize the structure of the procedure (Berthold et al., 2009).

In summary, self-explanation is an active or constructive learning activity that can improve learning outcomes in logical, rule-based domains. Self-explaining is effective whether it is initiated by the learner or a prompt, but learners need enough mental resources and relevant prior knowledge or enough support from the instructions to guide their explanations. Self-explanation prompts can provide additional information and guidance to help learners effectively self-explain, but they should provide only what is necessary to support the learner to avoid hindering explanations. Similarly, feedback on self-explanations can reduce floundering and incorrect explanations, but it can lead to overreliance on feedback. For procedural learning, rationale-based self-explanations help the learner develop a robust representation of a procedure and its applications.

Making rationale-based explanations is fundamentally the same as explaining the function of subgoals. Both explain how a set of steps contributes to the solution of a problem. Because self-explanation and subgoal learning both support effective organization of information and direct cognitive resources to structural features of worked examples, the present study explored whether self-explanation could provide an effective active or constructive method of learning subgoals. In addition, the present study explored whether subgoals created by an instructional designer could provide effective feedback to learners who are actively or constructively learning subgoals.

**1.4 Present Study**

The present study prompted participants to learn the subgoals of a procedure through a worked example that either encouraged passive, active, or constructive learning. The subgoals of the procedure were identified using the Task Analysis by Problem Solving (TAPS) procedure (Catrambone et al., 2016) that has been used in prior research (e.g., Margulieux & Catrambone, 2014). In the passive learning condition, participants were given subgoal labels created by the experimenters, as is conventional in prior subgoal research (e.g., Catrambone, 1998). These subgoal labels will also be created through the TAPS procedure (Catrambone et al., 2016).

In the active learning condition, participants were given the worked example grouped by subgoals and asked to select a subgoal label from a list of labels that matched the purpose of the group. The list contained only labels that were viable options, meaning the list did not include distracter items that were not applicable to the procedure being learned. Requiring novice learners to distinguish between explanations that might or might not apply to the procedure would likely have unnecessarily added to the cognitive load required to complete the task. Participants could have completed this activity incorrectly, though, by selecting the wrong label for a group of steps. This active method of self-explaining was equivalent to the active self-explanation methods used by Aleven and Koedinger (2002) and Conati and VanLehn (2000). The method matches Chi's (2009) definition of active learning as a method that requires activity from the learner but not construction of new information.

In the constructive learning conditions, participants were asked to create their own subgoal labels to explain the subgoals of the procedure. To train participants to construct their own labels, they were given subgoal label training (see Appendix A). Only the constructive groups received this training. The passive and active groups received a comparable task: analogy training (see Appendix B), as is common in the literature (Bielaczyc et al., 1995; Renkl et al., 1998). Non-constructive groups should not receive constructive training because it might prompt them to use constructive learning methods during the study, which could confound the results. Training for analogies (e.g., water : thirst :: food : hunger) was considered comparable because both analogies and subgoal labeling require people to consider the underlying relationship between words and come up with a new word that describes that relationship.

The three constructive learning conditions prompted participants to construct their own subgoal labels. They differed on the amount of guidance that participants received while constructing labels. There were two types of guided constructive conditions in which participants were given the worked example with the solution steps grouped by subgoal, and the example indicated which subgoals achieved the same functions. For instance, all of the subgoals denoted as “Label 1” achieve the same function though the contexts were different (see Figure 3).

| No labels | Given Labels (Passive) | Placeholder for Label (Constructive) |
|---|---|---|
| $4x - 8 = 2x + 6$<br>$+ 8 \qquad + 8$<br>$- 2x \qquad - 2x$<br>$4x - 2x = 6 + 8$<br>$2x = 14$<br>$/2 \quad /2$<br>$x = 7$ | $4x - 8 = 2x + 6$<br>Isolate variable<br>$+ 8 \qquad + 8$<br>$- 2x \qquad - 2x$<br>$4x - 2x = 6 + 8$<br>Simplify terms<br>$2x = 14$<br>$/2 \quad /2$<br>$x = 7$ | $4x - 8 = 2x + 6$<br>Label 1:___________<br>$+ 8 \qquad + 8$<br>$- 2x \qquad - 2x$<br>$4x - 2x = 6 + 8$<br>Label 2: __________<br>$2x = 14$<br>$/2 \quad /2$<br>$x = 7$ |

*Figure 3*. Worked example formatted with no labels, given labels, or placeholders for labels.

As is conventional in subgoal learning, participants saw multiple instances of each subgoal. Viewing multiple instances is critical to subgoal learning because it allows the learner to compare subgoals that achieve the same function but comprise different steps (Margulieux et al., 2012). In the guided constructive with hints condition, participants were given hints about the similarities among different instances of the same subgoal. In the guided constructive without hints condition, participant did not receive hints. In the unguided constructive condition, participants received a worked example that did not indicate which steps belonged to which subgoals. Participants in this condition had to identify the subgoals for themselves and create labels for them.

Schworm and Renkl (2006) argued that written self-explanations are better than spoken explanations because they are formal and recorded. They argued that writing explanations required articulating thoughts and creating a record that allowed students to reflect on their explanations more easily. They found that written self-explanations were the dominant predictor of improved learning outcomes over oral self-explanations. In the present study, self-explanations were written directly onto the worked example that participants studied.

The amount of guidance that participants received during instruction also differed based on whether they received feedback. Instructions for participants who received feedback had another copy of the worked example that included subgoal labels created by the experimenters. For the passive condition, this copy was exactly the same as the initial worked example. For the active and constructive conditions, the copied example with experimenter-created subgoal labels provided feedback to the participants about whether they selected the correct labels or created similar labels. Participants who received feedback were asked to compare their labels to those created by the experimenter to prompt them to reflect on the similarities or differences between the two. Instructions for participants who did not receive feedback included only the worked example with the passive, active, or constructive interventions. These participants were asked to re-read the example to make time on task more similar to that of participants who received feedback, as is common in the self-explanation literature (e.g., Chi et al., 1994). The exception was that participants in the passive and no feedback condition were not asked to re-read the example to make their experience different from those in the passive with feedback condition. Due to this difference, the time on task was different, providing some insight into whether time on task affects performance.

Because the worked example was long, participants received only one worked example. Giving one worked example provided a unique opportunity to ensure that participants in the feedback condition did not overly rely on feedback. Participants were not told that they would receive feedback until they completed the task, meaning that they did not know to expect feedback.

The guidance provided by feedback was expected to interact with subgoal learning methods. Learners who are making self-explanations can flounder because self-explanation, especially constructive explanations, requires some insight, meaning that learners have to recognize connections between pieces of information that are not necessarily apparent from the instructions (Wylie & Chi, 2014). Durso, Rea, and Dayton (1994) found that insight resulted from mental restructuring of knowledge that made connections between previously disjointed pieces of information. Durso et al. (1994) also found that if participants were given the solution to the problem at hand, mental restructuring did not occur.

Durso et al.’s (1994) findings can explain Schworm and Renkl’s (2006) and Wylie and Chi’s (2014) findings that learners who were given too much information did not benefit from self-explanation. Receiving information that could have been constructed through self-explanation does not allow for mental restructuring, which is one of the benefits of self-explanation (Wylie & Chi, 2014). Therefore, extra guidance from feedback on self-explanations was not always expected to lead to better learning outcomes, especially when learners received high levels of guidance during self-explanation. For instance, the constructive condition that explicitly draws connections between analogous subgoals was expected to promote a mental organization of information that fostered insight. Therefore, feedback in this condition was expected to be unnecessary or even detrimental. The constructive conditions that did not draw connections, however, were less likely to foster insight, and participants who did not have insights were expected to need feedback to learn the subgoals correctly. Though it is somewhat counterintuitive that more support could negatively impact learning, these expectations were similar to the expertise-reversal effect in which learners who are able to process information independently are hindered by instructional support (Sweller, 2010).

In the second experiment, the present study explored whether the knowledge constructed through self-explanation could be utilized to guide initial problem solving to improve later problem solving performance. In a review of 49 courses that used various instructional methods, Margulieux, McCracken, and Catrambone (2015) found that formative feedback from instructors during initial problem solving improved learning outcomes. This feedback tended to make abstract suggestions that were independent from the context of the problem. For instance, if a student was stuck on the first step of the problem in Figure 1, the instructor might prompt the student to isolate the variable that is being solved, but the instructor would not tell the student to subtract $2x$ from each side of the equation. Subgoal-oriented guidance during initial problem solving would provide the same type of context-independent support and might have the same effects.

Subgoal-oriented guidance in the second experiment scaffolded the practice problems with subgoal labels to guide initial problem solving. This method is similar to the problem completion effect in which solving partially solved problems improves later problem solving performance (Sweller, 2010). The scaffolding of practice problems differed between subjects. They either received unguided or one of two types of guided practice problems. The unguided practice problems gave participants a problem to solve and a blank space to solve it. For the guided practice problems, the blank space for solving the problem included subgoal labels that needed to be achieved to solve the problem (see Figure 4).

| **Unguided Practice Problem** | **Subgoal Labeled Practice Problem** |
| --- | --- |
| *Solve for x:*<br>4x – 8 = 2x + 6<br><br><br><br><br><br><br>*x* = | *Solve for x:*<br>4x – 8 = 2x + 6<br><u>Isolate variable</u><br><br><br><u>Simplify terms</u><br><br><br>*x* = |

*Figure 4.* Unguided practice problem compared to practice problem guided by subgoal labels.

The subgoal labels were either those created by the experimenters or those created by the participants. The labels created by the experimenter were expected to improve performance because they were similar to the abstract guidance that an instructor might provide. Participant-created labels were also expected to improve performance, if they were good labels, because self-explanation helps learners mentally organize information (Wylie & Chi, 2014). However, if participants struggled to make their own labels or did not trust that their explanations were correct, then participant-created labels were not expected to provide guidance during initial problem solving. Therefore, the effect that participant-created labels had on guiding initial problem solving was expected to provide information about the efficacy of the labels that participants constructed.

The problem solving domain for the present study was programming. Programming is a procedurally-focused STEM field that typically includes worked examples and practice problems in instruction. The acquisition of programming skill has been facilitated by self-explanation of goals and procedural structure (Soloway, 1986; Pirolli & Recker, 1994) and subgoal learning (Margulieux et al., 2012; Margulieux & Catrambone, 2014), so it was an appropriate domain for the interventions.

To control for prior knowledge, participants were required to have little programming knowledge. Because participants were novices, the present study used a drag-and-drop programming language to teach programming concepts. Drag-and-drop programming languages are more easily understood by novice learners because they can select and drag pieces of code from a menu, which does not require learning the syntax and semantics of a programming language (Hundhausen, Farley, & Brown, 2009; see Figure 5). The programming language used in the present study was Android App Inventor, which is used to create applications (apps) for Android devices. Participants used App Inventor to create an app that has buttons that play sounds when pressed.

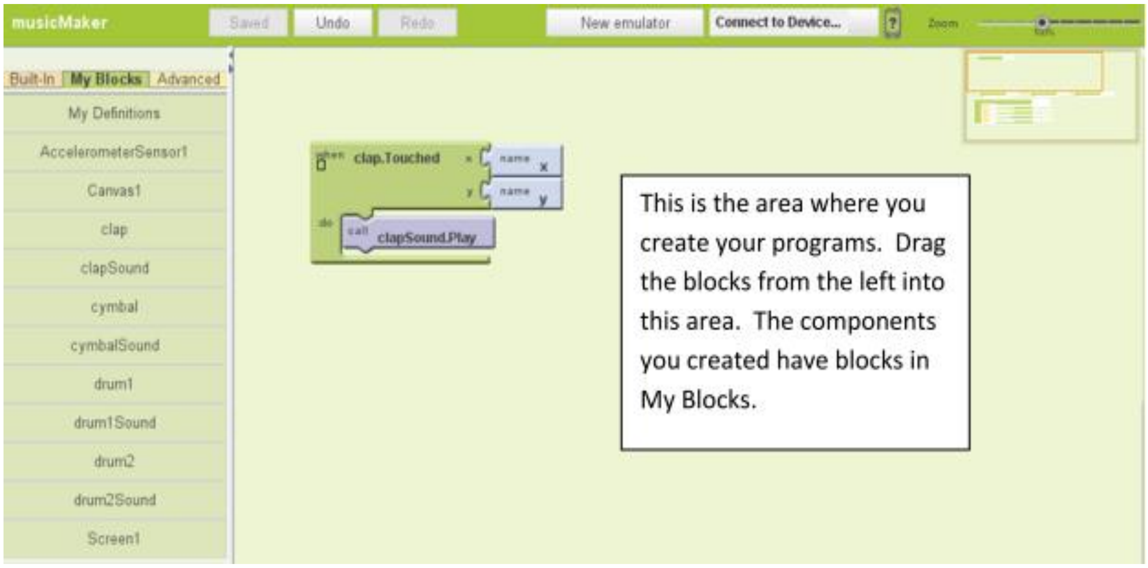


*Figure 5*. App Inventor interface with interlocking pieces of code selected from menus used to program features.

# CHAPTER 2: EXPERIMENT 1

Experiment 1 explored the efficacy of various methods of learning subgoals. It manipulated whether participants passively, actively, or constructively engaged with subgoal labels and whether participants received feedback while they were learning.

## 2.1 Method

**2.1.1 Participants.** Each of the 10 conditions had 20 participants ($N$ = 200). Participants were students at the Georgia Institute of Technology and recruited through the SONA experimental system and announcements in psychology classes. To qualify for participation in the experiment, a person must not have had experience with Android App Inventor and must not have taken more than one high school or college-level course in computer science or computer programming. These limitations were necessary because instructional materials were designed for novices.

**2.1.2 Pre-instruction procedure and materials.** Sessions took between 80 and 110 minutes, depending on how quickly participants complete each of the tasks. First, participants completed the demographic questionnaire, working memory measure, and pre-test, which took 10 to 15 minutes. Demographic information was collected for participants' age, gender, academic field of study, high school GPA, college GPA, year in school, computer science experience, comfort with computers, and expected difficulty of learning App Inventor because they are possible predictors of performance (Rountree, Rountree, Robins, & Hannah, 2004; see Table 1). These demographic characteristics were not found to correlate with problem solving performance (see Table 1).

Table 1. *Demographic Averages for Participants and Their Correlation with Problem Solving Performance in Experiment 1.*

| | Averages | | Correlation | |
|---|---|---|---|---|
| | *M* | *SD* | *r* | *p* |
| Gender | 55% male | - | .12 | .10 |
| Age | 19.5 | 2.3 | -.09 | .23 |
| High School GPA | 3.88 | .24 | -.03 | .72 |
| Year in College | 1.99 | 1.3 | .03 | .69 |
| College GPA | 3.40 | .48 | .13 | .09 |
| Comfort with Computers (out of 7) | 4.09 | 1.6 | .12 | .09 |
| Expected Difficulty (out of 7) | 4.15 | 1.3 | .13 | .08 |
| Previous CS Courses | 45% taken 1 course | - | .03 | .68 |

Participants' working memory capacity was measured because previous research has found that working memory capacity predicts success at self-explanation (Wylie & Chi, 2014). The Shapebuilder task was used to measure working memory capacity (Atkins et al., 2014). The Shapebuilder task is a four-dimensional task that includes a four-by-four grid and four sets of shapes (i.e., square, circle, diamond, and triangle) in different colors. The task presents to the participant a sequence of colored shapes on the grid, and the participant is asked to match the order, location, shape, and color of the items presented (Atkins et al., 2014). This task is similar to the problem solving procedure of creating an app because both involve dragging items of various shapes and colors in a particular order to correctly achieve the task; therefore, it was considered an appropriate tool for measuring working memory capacity for this procedure.

Participants completed a multiple-choice pre-test to ensure that they did not have prior knowledge of the procedure. The five pre-test questions asked about the most basic App Inventor features to capture any rudimentary knowledge that participants had about the procedure. For each question in the pre-test, one of the answer choices was “I don’t know” to avoid forcing participants to guess and introducing unnecessary error. The majority of participants (91%) scored a zero on the pre-test, and no participants scored higher than one point.

**2.1.3 Instructional procedure and materials.** After the pre-instruction period, participants started the instructional period, which took 40 to 55 minutes. All manipulations occurred within the instructional period. The instructional period started with an overview video of the App Inventor interface that was the same across all participants. The purpose of this video was to introduce participants to the App Inventor interface and the types of tasks that can be completed with App Inventor. The video did not include information about the procedure being taught, but it was intended to help participants familiarize themselves with the problem solving space in which they would be working. Palmiter, Elkerton, and Baggett (1991) suggested that videos are a medium that help participants intuitively learn a direct manipulation interfaces, like App Inventor.

After the introductory video, participants received either subgoal label or analogy training (Appendices A and B, respectively). Participants who constructed subgoal labels received subgoal label training, and those who did not construct labels received analogy training. Next, participants received the worked example. The worked example listed the steps taken to create a Music Maker app that plays musical sounds when images of instruments are pressed or the device is shaken. For instance, a drum sound would play when a drum image is pressed, or a tambourine sound would play when the phone is shaken. The format of the worked example depended on participants' assigned method of subgoal learning. The passive method gave participants subgoal labels (see passive condition in Figure 3), and the other methods gave participants spaces to fill in subgoal labels (see constructive condition in Figure 3), except for the unguided constructive condition, which had only the listed steps. For the active method, participants had a word bank with labels that they could select. For the constructive methods, participants did not have a word bank. In the guided constructive with hints condition, additional text highlighted similarities between all subgoals called "Label 1." This guidance was given for each subgoal.

When participants finished the first pass through the worked example, they were either prompted to re-read the example for the no feedback condition, or they were given the worked example with the experimenter-created subgoal labels for the feedback condition. Participants in the feedback condition were told that the subgoal labels in the second copy of the worked example were created by subgoal label experts. Then they were asked to compare the labels that they made or selected to those given in the second example. For the passive condition, the second copy of the example was exactly the same as the first copy. Therefore, it did not provide feedback, per say, but the passive group did not need feedback because it did not select or construct labels. To make the passive no feedback and passive feedback conditions different, participants in the no feedback condition were not asked to re-read the example. This difference provided some insight into the effect of re-reading the example and time on task.

To ensure that participants paid attention to the worked example and could complete tasks in the App Inventor interface, they were asked to complete practice problems before finishing the instructional period. Of the four tasks that participants completed, two required isomorphic transfer from the worked example, meaning that they used the same context and procedural steps as the worked example and differed only in surface features. For instance, the worked example showed the steps to create a drum image that plays a drum sound when pressed, and an isomorphic transfer practice problem asked participants to create a cymbal image that plays a cymbal sound when pressed. The other two practice problems required contextual transfer, meaning that they used the same procedural steps as the worked example and differed in surface and contextual features. Continuing the prior instance, a contextual transfer practice problem asked participants to create a button that changes color when pressed.

**2.1.4 Assessment procedure and materials.** To measure cognitive load experienced during the instructional period, participants completed a questionnaire for measuring cognitive load induced during programming instruction that was developed by Morrison, Dorn, and Guzdial (2014). This questionnaire was given directly after the instructional period. The questionnaire included three questions about intrinsic cognitive load (i.e., the load associated with processing information that is necessary to learn the procedure), three questions about extraneous cognitive load (i.e., the load associated with processing information that is not necessary to learn the procedure), and four questions about germane cognitive load (i.e., the load associated with cognitive processes of learning).

Following the cognitive load questionnaire, participants took a post-test that contained the same items as the pre-test. The post-test served as a learning check because the questions were about the most basic features of App Inventor. The majority of participants (82%) scored the full five points on this post-test, and no participants scored lower than four points. A score lower than four points on the post-test would suggest that a participant did not pay adequate attention to the instructions and should be removed from the analyses. No participants met this criterion, and, therefore, all participants were included in the analyses. Participants were also asked to rate how well they understood the instructions and how comfortable they would be solving novel problems using the procedure.

After these checks, participants completed assessment tasks that measured learning. During this assessment period, participant did not have access to the instructional materials. They were told of this restriction at the beginning of the session. The first set of assessment tasks was problem solving tasks that asked participants to modify or add components to their Music Maker app. Of the five tasks, two required contextual transfer from the worked example, meaning that the superficial features of the app components were different but the procedural steps used to create them were the same. The remaining three tasks required procedural transfer from the worked example, meaning that the individual steps used to create the app components were different but the procedure used to create them was structurally the same. For instance, the worked example showed steps to make a sound play when an image is clicked, and a problem solving task asked participants to make a label display text when an image is clicked. The procedure used to create these two features was the same, but the steps taken to do so were different.

Participants were asked to attempt the tasks in the App Inventor interface and then to write down the steps that they took so that their problem solving process could be scored. In the interface, completing later steps of a task can rely on correctly completing earlier steps of a task. For instance, a participant could not program a sound to play when an image is clicked if they could not create the image. For this reason, participants were asked to write steps that they would take to complete the task, even if they could not complete them in the interface. Participants had up to 25 minutes to complete the problem solving tasks.

The second set of assessment tasks was explanation tasks intended to measure whether participants could recognize the function that a step of a solution serves. Participants received solutions to problem solving tasks and were asked to match each step of the solution to the subgoal label that correctly explains the function of that step. Participants were given the solutions to the problem solving tasks that they had just attempted to solve to make the problem solving and explanation tasks more congruous and reduce the amount of new, superficial information that participants needed to process.

**2.1.5 Design.** Experiment 1 was five-by-two factorial, between-subjects design: subgoal learning method (passive, active, guided constructive with hints, guided constructive without hints, or unguided constructive) was crossed with feedback (no feedback or feedback). Dependent measures were performance on the problem solving tasks, performance on the explanation tasks, and time on task for the assessments and for the instructional period. Demographic characteristics, working memory capacity, pre-test and post-test score, subjective cognitive load, and perception of understanding were also collected as possible predictors of performance. The subgoal labels that participants construct were collected and analyzed for content. Quality of subgoal label was also considered as a possible predictor of performance.

## 2.2 Results and Discussion

### 2.2.1 Guided constructive learning improved problem solving performance.

For the problem solving assessment, participants received a score for number of correct steps taken towards problem solutions. Because the tasks involve multiple steps, scoring based on steps rather than whole answers provided more sensitivity. The maximum possible score was 25. Performance on the problem solving tasks depended on the interaction of subgoal learning method and feedback, $F(4, 190) = 3.39$, $MSE = 23.6$, $p = .01$, partial $\eta^2 = .067$ (see Figure 6). Due to the disordinal nature of this interaction, the main effects will not be reported to avoid confusion in interpreting the results (Maxwell & Delaney, 2004). To explore this interaction and determine the effect of feedback on each method of learning subgoals, simple main effects comparisons were used. This analysis found that feedback affected performance only for the guided constructive groups, but it affected them in different ways (see Table 2). Participants in the guided constructive with hints conditions performed statistically better when they did not receive feedback than when they did, whereas participants in the guided constructive without hints conditions performed statistically better when they received feedback than when they did not.

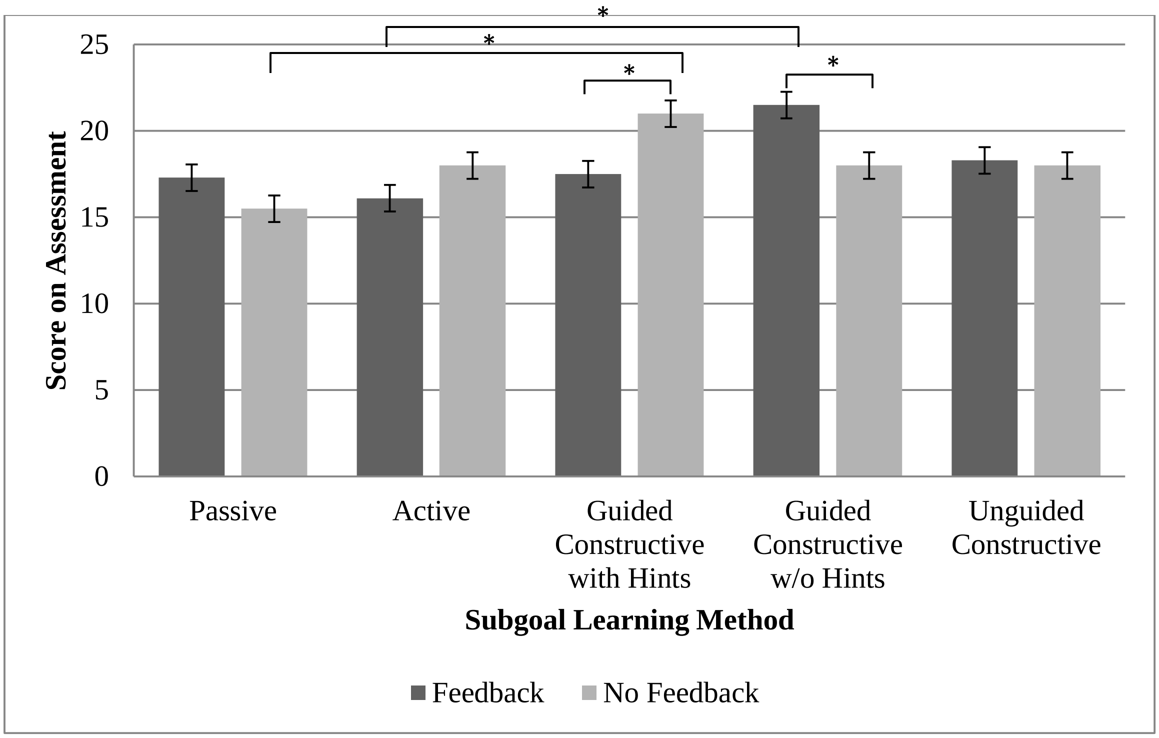


*Figure 6*. Performance on problem solving tasks among conditions in Experiment 1. Maximum possible score was 25. Error bars are standard error. Statistically significant differences are indicated with asterisks.

Table 2. *Simple Main Effects Analysis of Subgoal Learning Methods on Problem Solving Performance in Experiment 1.*

| Subgoal Learning Method | Mean for No Feedback | Mean for Feedback | Mean Difference | Std. Error | *p* |
|---|---|---|---|---|---|
| Passive | 15.5 | 17.4 | -1.90 | 1.54 | .218 |
| Active | 18.0 | 16.1 | 1.95 | 1.54 | .206 |
| Guided Constructive with Hints | 21.0 | 17.5 | 3.50 | 1.54 | .024 |
| Guided Constructive without Hints | 18.0 | 21.5 | -3.54 | 1.54 | .023 |
| Unguided Constructive | 18.0 | 18.3 | -.30 | 1.54 | .845 |

To explore the relative efficacy of different methods of learning subgoals, a simple main effects comparison was used for the feedback variable. The method of learning subgoals affected performance for groups that received feedback, $F(4, 190) = 3.54$, $MSE = 23.6$, $p = .008$, partial $\eta^2 = .069$, and groups that did not receive feedback, $F(4, 190) = 3.27$, $MSE = 23.6$, $p = .013$, partial $\eta^2 = .064$. Based on pairwise comparisons within the two types of feedback groups, for participants who did not receive feedback, those in the guided constructive with hints condition performed statistically better than those in the passive condition, Mean Difference = 5.55, $p = .004$. Furthermore, for participants who received feedback, participants in the guided constructive without hints condition performed statistically better than those in the active condition, Mean Difference = 5.45, $p = .005$. These results suggest that, within both the feedback and no feedback groups, the best performing conditions scored statistically significantly better than those in the worst performing conditions. The other conditions that scored in the middle were not statistically better or worse than the best or worst performing conditions.

This pattern of results matched the expected pattern of results well, providing support for the hypothesis that there is an optimal level of support for learning subgoals. In particular, the disordinal effect of feedback on the guided constructive groups suggests that learners perform best with just enough support and providing too much support hindered learning. In addition, passive and active methods of learning subgoals produced the worst results, likely because they provided too much support. Based on these results, it was concluded that providing hints for learners constructing subgoal labels and providing feedback on constructed labels are both techniques that can help learners to perform better on later problem solving, but providing both types of support could hurt performance. On the other end of the support spectrum, the unguided constructive conditions performed in the middle of all the conditions. Perhaps for some students, this low level of support was ideal while it was too little support for others. To address this question, individual differences in working memory and quality of subgoal labels were explored.

An ANCOVA was used to determine whether working memory capacity was a covariate of the interventions' effect on problem solving performance. Working memory capacity was not found to affect problem solving performance, $F(1, 190) = .47$, $MSE = 23.6$, $p = .492$, partial $\eta^2 = .003$; therefore, the interaction effect of method of learning subgoals and feedback on problem solving performance did not depend on individual differences in working memory.

**2.2.2 Analysis of subgoal label quality.** To determine the quality of participant-created labels, they were analyzed. Each label was analyzed as one unit (i.e., each word within a label was not analyzed individually), and each participant was categorized based on all of their labels collectively. In nearly all cases, all of the labels that a participant created fell into one of the following categories. The coding scheme that was determined *a priori* included categories for whether labels were context-specific, context-independent, or incorrect. Two raters scored 20% of participants and compared their scores. Interrater reliability was measured with intra-class correlation coefficient of agreement because the scale of measurement for categories was nominal and absolute agreement was necessary. Reliability was high, ICC(A) = .98, and the remaining 80% of participants were scored by a single rater.

Context-specific labels included information about the specific instantiation of the subgoal and, therefore, could be applied only to that one instantiation. For example, the participant-created label “name and add picture to image sprite” could be applied only to the steps that named and added a picture to an Image Sprite. For a participant to be classified as context-specific, at least 80% of labels had to include information about the context. In all cases except two, participant labels were either completely context-specific or completely context-independent.

Context-independent labels, on the other hand, did not contain any information about the specific instantiation of that subgoal. For example, the participant-created label “add properties to app” is context-independent because it can be applied to any property, such as the name and picture of an Image Sprite, that is being added to the app. For a participant to be classified as context-independent, at least 80% of labels had to not include information about the context. Context-independent labels were considered to be of a higher quality than context-specific labels because they indicate a more conceptual understanding of the procedure that is more easily applied to solving new problems. Because context-specific labels include information about the context of the current problem, they cannot be applied directly to novel problems.

Incorrect subgoal labels were those that were execution-based instead of function-based, such as “click on menu,” or those that did not describe the correct function. For a participant to be classified as incorrect, more than one label had to meet either of these criteria. In all cases except one, for participants who made incorrect labels, at least 80% of their labels were incorrect.

While implementing this coding scheme, two more categories were defined. For the guided constructive with hints conditions, many of the constructed labels included terms from the hints. For example, the hint for the subgoal that defines the output of an interaction included the term “output,” and many participants who received hints included the term “output” in the labels that they created. In all cases, participant-created labels that used terms from the hints were context-independent. To distinguish these labels from the other context-independent labels, these labels were classified as hint-term context-independent labels. For a participant to be classified as hint-term context-independent, at least three out of the five labels had to include terms from the hints. If fewer than three labels included terms from the hints, then the participants was classified as context-independent.

For the unguided constructive conditions, many of the subgoals that participants identified for themselves included many more steps than the subgoals created by experimenters. For example, some subgoals that participants grouped were more than 20 steps long, whereas the longest experimenter-grouped subgoal was seven steps. In all cases, the participant-created labels for these higher level subgoals were context-specific. For example, one participant identified a subgoal that was 24 steps long and labeled it “make the correct sounds play according to whatever input is received.” To distinguish these labels from the other context-specific labels, these labels were classified as higher-level context-specific labels. For a participant to be classified as higher-level context-specific, the participant-identified subgoals had to include at least twice as many steps the subgoals identified by experimenters because in these cases, participants were lumping two or more experimenter-identified subgoals together. The higher-level context-specific labels were considered lower quality subgoal labels than the context-independent or context-specific labels. One of the benefits of learning the subgoals of a procedure is that subgoals break up long procedures into functional pieces that are easier to adapt to novel problems. The higher-level subgoals did not identify these functional pieces but instead described the procedure that was being executed. Describing the procedure in this way instead of in a functional way is theoretically less conducive to transfer to novel problems.

**2.2.4 Hint improved participant-created subgoal label quality.** For examples of participant-created labels for all five classifications, see Table 3. The majority of context-independent subgoal labels (91%) were two to five words long, which is similar to the experimenter-created labels. The context-specific labels, on the other hand, tended to be longer – 54% were longer than five words – because they included context-specific words, such as specifically mentioning the drum sound.

Many participants in the guided constructive with hints conditions created hint-term context-independent labels (45%) or context-independent labels (24%). A smaller number of these participants created context-specific labels (22%) or incorrect labels (8%). Many participants in the guided constructive without hints conditions created context-independent labels (49%). A proportion of these participants created context-specific labels (27%) or incorrect labels (24%). The majority of participants in the unguided constructive conditions created higher-level context-specific labels (79%). A small number of these participants created context-independent labels (9%), context-specific labels (9%), or incorrect labels (3%).

Table 3. *Examples of Subgoal Labels Constructed by Participants for Each of the Coding Classifications in Experiment 1.*

| Experimenter-Created Label | Context-Specific | Higher-Level Context-Specific | Context-Independent | Hint-Term Context-Independent | Incorrect |
|---|---|---|---|---|---|
| Create component | Create image sprite | Create a canvas that fills the screen | Add component to app | Begin new object | Define variable |
| Set properties | Name and add picture to image sprite | | Edit component | Add properties to app | Select/drag |
| Handle input | Add condition for when clap is touched | Make a sound when clap icon is touched | Add interface command | Set user inputs | Program functions |
| Set output | Make clapsound play when clap is touched | | Set command result | Set outcomes of inputs | Specify function |
| Set conditions | Make something happen if the user moves the phone | When the phone is tilted down, the clap sound will play | Add command conditions | Establish input conditions | New function |

To determine whether the type of subgoal labels that participants made affected problem solving performance, a Kruskall-Wallis *H* test was used. The *H* test was deemed more appropriate than the *F* test for this analysis because the number of participants in each group (i.e., type of subgoal labels) was not equal, violating one of the assumptions of the *F* test. The type of subgoal labels that participants created was also a quasi-experimental variable, making a non-parametric test more valid. The limitation of the *H* test, however, is that it is more conservative than the *F* test. The *H* test was not statistically significant, $p = .17$, though the median scores (reported here instead of means because the *H* test uses median scores) were higher for context-independent labels ($M$ for context-independent hint-term = 21.1, $M$ for context-independent = 19.5) than for context-specific labels ($M$ for higher-level context-specific = 18.2, $M$ for context-specific = 17.8). The average standard deviation for these groups was 4.87, making the error too large to find statistically significant differences between groups. Perhaps with a larger sample size or a less conservative test, this finding would be statistically significant.

Though the types of labels that participants created were not found to directly affect problem solving performance, most of the participants in the guided constructive with hints conditions created subgoal labels that were very similar to the experimenter-created labels, meaning that they created labels that aligned with those created through an intensive task analysis with a subject-matter expert. For this reason, these participant-created labels were considered to be high quality subgoal labels. Because these conditions led to high-quality labels, it is not surprising that participants performed better on the problem solving tasks when they did not receive feedback (i.e., experimenter-created labels) compared to when they did receive feedback. Because participants created high quality labels, comparing their labels to the experimenter-created labels in the feedback might not have been as beneficial as reviewing the labels that they constructed, as participants in the no feedback condition did. Comparing labels might have caused participants to unjustifiably question or doubt their understanding of the procedure, whereas reviewing their own labels would reinforce the mental representations that participants developed. This effect is similar to the expertise-reversal effect in which giving instructional support to students helps their learning if they have a low level of prior knowledge but hinders their learning if they have a high level of prior knowledge (Sweller, 2010).

Participants in the guided constructive without hints conditions made more context-specific or incorrect labels (51%) than those who received hints (30%). Therefore, on average these participants had lower quality labels than those who received hints. This difference can explain why participants who did not receive hints performed better when they received feedback than when they did not. The feedback likely provided necessary support for these participants, improving their problem solving performance.

Most participants in the unguided constructive conditions grouped subgoals that were different than the subgoals identified by the experimenter and labeled these subgoals with context-specific labels. Because these labels were different from the experimenter-created labels in multiple aspects, it is not surprising that feedback did not affect performance for the unguided constructive conditions. The feedback likely provided guidance that was so different from the participants' mental representations of the procedure that they could not reconcile the two different representations.

Participants in the unguided constructive condition also spent much more time looking at the feedback that those in other conditions. A main effect of subgoal learning method was found for time spent looking at feedback, $F(4, 190) = 6.97$, $MSE = 2.23$, $p < .001$, partial $\eta^2 = .140$ (see Figure 7). Using Tukey's HSD post-hoc analysis, the unguided constructive group was the only group found to have a statistically significant mean difference from the passive group (Mean Difference = 1.30, $p = .015$), active group (Mean Difference = 1.59, $p < .001$), guided constructive with hints (Mean Difference = 1.32, $p = .001$), and guided constructive without hints (Mean Difference = 1.29, $p = .001$). There was no main effect of feedback condition on feedback time, $F(1, 190) = .03$, $MSE = 2.23$, $p = .86$, partial $\eta^2 < .00$, or interaction of subgoal learning method and feedback, $F(4, 190) = .31$, $MSE = 2.23$, $p = .82$, partial $\eta^2 = .005$. These time on task results further suggest that participants in the unguided constructive conditions had difficulty reconciling the labels that they created with those presented in the feedback, making the experimenter-created labels a poor source of feedback for this group.

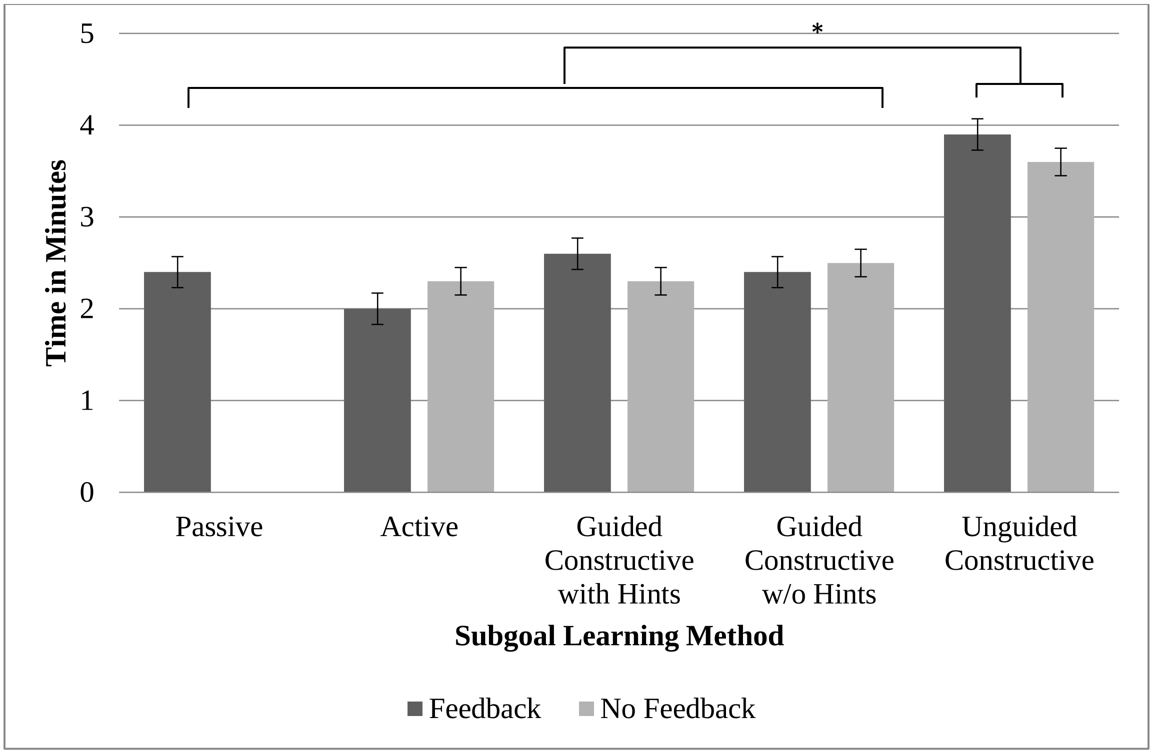


*Figure 7.* Time spent on reviewing feedback among conditions in Experiment 1. Error bars are standard error. Statistically significant differences are indicated with asterisks.

**2.2.5 More engaging methods of learning increased instructional time.** Time that participants spent on each part of the experimental session was collected. There were differences among groups for time spent on the worked example and time spent working on practice problems. For time spent on the worked example, which includes using the worked example to re-create the app and learning the subgoals of the procedure through passive, active, or constructive methods, there was a main effect of subgoal learning method, $F(4, 190) = 19.37$, $MSE = 34.2$, $p < .001$, partial $\eta^2 = .29$ (see Figure 8).

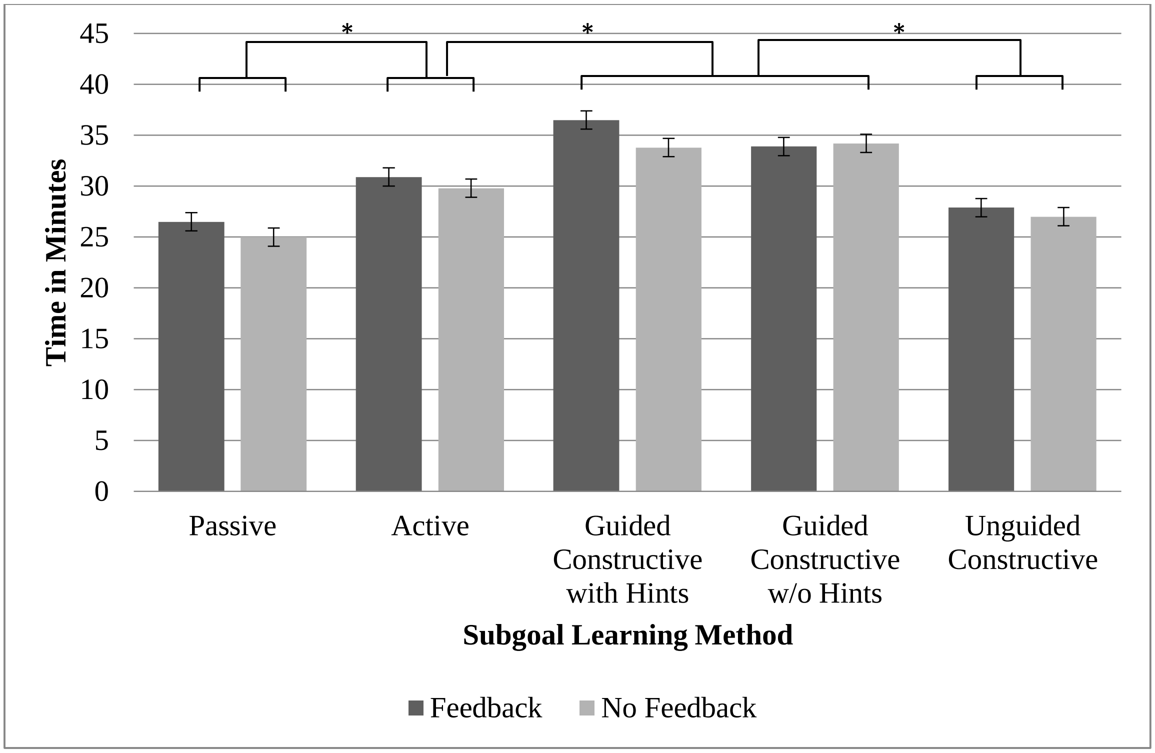


*Figure 8.* Time spent using the worked example, including re-creating the app and engaging in subgoal learning methods, among conditions in Experiment 1. Error bars are standard error. Statistically significant differences are indicated with asterisks.

The passive ($M$ = 25.8 minutes, $SD$ = 6.0) and unguided constructive ($M$ = 27.4 minutes, $SD$ = 6.2) groups completed this part of the instructional period quickest and were not statistically different from each other (Mean Difference = 1.68, $p$ = .70). The active group ($M$ = 30.4 minutes, $SD$ = 5.3) took statistically significantly longer than the passive group (Mean Difference = 4.59, $p$ = .005) but not the unguided constructive group (Mean Difference = 2.91, $p$ = .17). The guided constructive groups took statistically longer than the active group (with hints, Mean Difference = 4.82, $p$ = .003; without hints, Mean Difference = 3.69, $p$ = .041) and were not statistically different from each other (Mean Difference = 1.13, $p$ = .91).

With the exception of the unguided constructive group, the more engaging methods of learning subgoals took longer to complete than the less engaging methods. These results were expected because higher levels of engagement take more thought and, therefore, time to complete. The unguided constructive group might have taken less time because participants tended to construct high-level subgoal labels that described the process of creating the Music Maker app instead of the conceptual procedure for creating apps. This level of description is much easier to identify than a deeper, conceptual description. There was no main effect of feedback on time spent using the worked example, $F(1, 190) = 1.41$, $MSE = 34.2$, $p = .21$, partial $\eta^2 = .03$, or interaction of subgoal learning method and feedback, $F(4, 190) = .35$, $MSE = 34.2$, $p = .85$, partial $\eta^2 = .01$. Whether participants received feedback did not affect the time they spent using the worked examples. This result was expected because this measurement was taken before participants knew that they would receive feedback.

**2.2.6 Feedback increased time spent on practice problems but not assessments.** For time spent on practice problems, a main effect of feedback was found, $F(1, 190) = 7.92$, $MSE = 9.8$, $p = .005$, partial $\eta^2 = .04$ (see Figure 9).

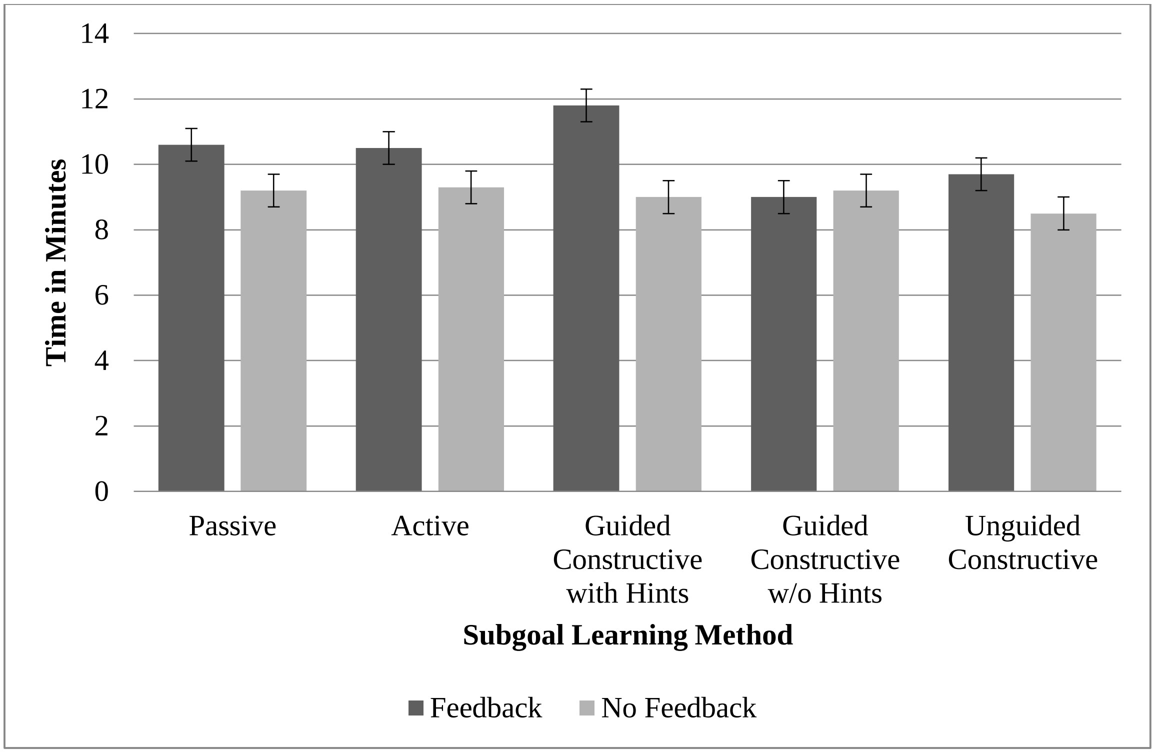


*Figure 9.* Time spent working on practice problems among conditions in Experiment 1. Error bars are standard error.

Participants who received feedback (*M* = 10.3 minutes, *SD* = 3.3) spent an extra 14% on solving practice problems than those who did not (*M* = 9.1 minutes, *SD* = 3.0). This effect might be due to participants referencing both the worked example and feedback while solving practice problems instead of referencing only the worked example. The effect accounts for only 4% of the variance in time, however, so the effect of feedback on practice problem time is small. There was no main effect of subgoal learning method on time spent working on practice problems, $F(4, 190) = 1.26$, $MSE = 9.8$, $p = .29$, partial $\eta^2 = .03$, or interaction of subgoal learning method and feedback, $F(4, 190) = 1.14$, $MSE = 9.8$, $p = .34$, partial $\eta^2 = .02$. Method of subgoal learning, therefore, did not affect the time participants spent working on practice problems.

The last time measurement was time spent on problem solving tasks. Participants spent an average of 23.40 minutes on the problem solving tasks (*SD* = 3.0). No differences among conditions were found for this measurement. There was no main effect of subgoal learning method, $F(4, 190) = 1.51$, $MSE = 9.0$, $p = .20$, partial $\eta^2 = .03$, no main effect of feedback, $F(1, 190) = 1.97$, $MSE = 9.0$, $p = .16$, partial $\eta^2 = .01$, and no interaction of method and feedback, $F(4, 190) = .81$, $MSE = 9.0$, $p = .52$, partial $\eta^2 = .02$. Based on these results, the interventions did not affect the time it took participants to complete the problem solving tasks. Groups that performed better or worse on problem solving performance did not differ on the amount of time that it took to solve problems.

**2.2.7 No differences found in other metrics.** For the explanation assessment, participants received a point for each step that was correctly paired with its functional label. The maximum possible score was 20. The mean score on this assessment for all groups was 15.24 with a standard deviation of 4.66. No statistical differences were found for performance on the explanation task among the conditions. There was no main effect of subgoal learning method, $F(4, 190) = 1.51$, $MSE = 21.9$, $p = .20$, partial $\eta^2 = .037$, no main effect of feedback, $F(1, 190) = .64$, $MSE = 21.9$, $p = .42$, partial $\eta^2 = .004$, and no interaction, $F(4, 190) = .30$, $MSE = 21.9$, $p = .88$, partial $\eta^2 = .008$. These results suggest that participants in all conditions were equally prepared to complete the explanation task, regardless of whether they had seen the experimenter-created labels in the instructions or not. Because participants could match the functions of subgoals to the experimenter-created labels even if they had not seen the labels before, this finding suggests that participants could equally recognize the correct experimenter-created label that matched subgoals' functions.

At the end of the instructional period, including worked example, feedback or review, and practice problems, participants were asked to rate their cognitive load while learning the procedure. This measure was intended to assess whether there were significant cognitive load differences between the conditions that might affect participants' experience of learning. Overall self-report of cognitive load was not affected by the method of subgoal learning, $F(4, 190) = 1.64$, $MSE = 163.72$, $p = .17$, presence of feedback, $F(1, 190) = .30$, $MSE = 163.72$, $p = .58$, or their interaction, $F(4, 190) = 1.59$, $MSE = 163.72$, $p = .18$ (see Figure 10).

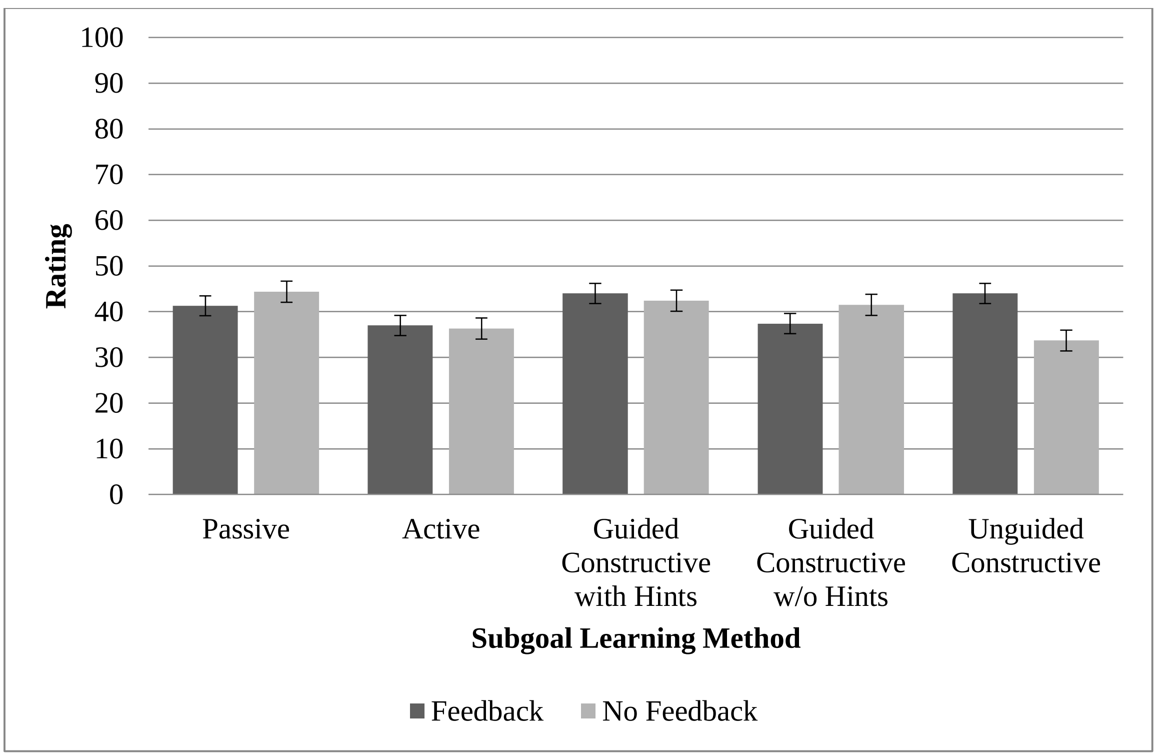


*Figure 10.* Self-reported rating of cognitive load while working through the instructional period in Experiment 1. Error bars are standard error.

In addition, no differences were found within each of the three types of cognitive load: intrinsic, extraneous, and germane (see Table 4). These results suggest that participants did not perceive differences in cognitive load among the conditions; therefore, the participants constructing labels did not perceive a higher cognitive load than participants performing less engaging tasks. This finding is important because if constructing labels was more cognitively taxing, learners might be less inclined to do it even if it improves learning. These results suggest that this was not the case.

Table 4. *ANOVA Results for Intrinsic, Extraneous, and Germane Cognitive Load Measures for Experiment 1.*

| | Main Effect of Subgoal Learning Method | | Main Effect of Feedback | | Interaction | |
|---|---|---|---|---|---|---|
| | *F* | *p* | *F* | *p* | *F* | *p* |
| Intrinsic Load | .83 | .51 | .27 | .60 | .35 | .84 |
| Extraneous Load | .15 | .96 | .13 | .97 | .74 | .57 |
| Germane Load | 1.77 | .14 | .20 | .66 | 2.13 | .08 |

Participants were asked to rate how well they understood the instructions from "1 – Not well at all" to "7 – Very well." In general, participants rated that they understood the instructions well ($M$ = 5.98, SD = 1.1). These ratings were not affected by the method of subgoal learning, $F(4, 190) = .29$, $MSE = 1.13$, $p = .89$, presence of feedback, $F(1, 190) = 2.03$, $MSE = 1.13$, $p = .16$, or their interaction, $F(4, 190) = 1.08$, $MSE = 1.13$, $p =$

.37. Participants were also asked to rate how comfortable they were solving novel problems from “1 – Not comfortable at all” to “7 – Very comfortable.” Participants rated that they were comfortable solving new problems ($M$ = 5.52, SD = 1.3). These ratings were not predicted by the method of subgoal learning, $F(4, 190) = 1.02$, $MSE = 1.57$, $p = .40$, presence of feedback, $F(1, 190) = 1.02$, $MSE = 1.57$, $p = .31$, or their interaction, $F(4, 190) = 1.90$, $MSE = 1.57$, $p = .11$. These results indicate that participants in different conditions felt equally prepared to solve novel problems, even though some of them performed better than others. Because perceived understanding and comfort solving novel problems were equivalent across groups in this study, these factors were not expected to have affected participants’ problem solving performance.

In summary, Experiment 1 explored the tradeoffs between instructional guidance and constructing knowledge for learning a procedure. The results suggest that constructive methods of learning subgoals are the most effective, but they require some instructional support. Either receiving feedback on constructed labels or receiving hints while constructing labels, but not both, led to the best problem solving performance. Participants who received hints while constructing labels were more likely to construct high quality labels than participants who did not receive hints. These participants performed better when they did not receive feedback than when they did, suggesting that, for those who received hints, the feedback provided too much additional instructional support to promote constructive learning. In contrast, participants who did not receive hints performed better when they received feedback than when they did not, suggesting that the feedback was necessary for the best performance when participants did not receive hints.

# CHAPTER 3: EXPERIMENT 2

The problem completion effect states that learners perform better on later problem solving when they receive more guidance during initial problem solving (Sweller, 2010). This experiment tested whether subgoal guidance during practice problems could further improve problem solving performance. Specifically, Experiment 2 explored whether participant-created labels could effectively scaffold initial problem solving. Participants guided by their own labels were compared to participants without guidance (i.e., equivalent to Experiment 1) and participants guided by experimenter-created labels. If participant-created labels represented well-organized knowledge about the procedure, then participants who received practice problems scaffolded with their own labels should have performed better than those did not receive guidance. Based on the results of Experiment 1 in which experimenter-created labels did not serve as effective feedback, scaffolding with experimenter-created labels was not expected to improved later problem solving performance.

## 3.1 Method

The procedure used in Experiment 2 was almost identical to that used in Experiment 1, but the manipulation was different. Experiment 1 manipulated the worked example, and Experiment 2 manipulated the practice problems. The practice problems in Experiment 2 were either unguided (i.e., as they were in Experiment 1), guided by experimenter-created subgoal labels, or guided by participant-created subgoal labels. All other aspects of the method, including sample size per condition ($n$ = 20, $N$ = 60), procedure, and measurements were the same.

For the instructions in Experiment 2, one of the two most effective conditions from Experiment 1 was used: the guided constructive with hints and without feedback condition. This condition was chosen over the guided constructive without hints and with feedback condition because the feedback aspect requires that an instructional designer develop subgoal labels for the worked example. If an instructional designer does not have to create subgoal labels, then the constructive subgoal intervention requires fewer resources to apply to instructions in the future, making it the more pragmatic technique for improving learning. In addition, not providing feedback to participants means that participants' first exposure to the experimenter-created labels will be as scaffolding in the practice problems. The experimenter-created labels did not provide effective feedback to participants in the guided constructive with hints condition in Experiment 1, and using the experimenter-created labels as scaffolding allows further exploration of their efficacy.

The participants in Experiment 2 had similar demographic characteristics as those in Experiment 1 (see Table 5). The majority of participants (85%) scored zero points on the pre-test, and the remaining participants scored one point out of the possible five. After the instructional period, the majority of participants (90%) scored the full five points, and the remaining participants scored four out of five points, suggesting that all participants paid attention to the instructions. No participants were removed from analysis.

### 3.2 Results and Discussion

The condition from Experiment 2 that received unguided practice problems had the exact same instructions as the guided constructive with hints and without feedback condition from Experiment 1. Both conditions received the guided constructive with hints worked example and unguided practice problems. The means from this condition in both experiments were compared to ensure that the participants in each experiment performed equivalently. Participants in these two groups performed similarly, and the means were within the margin of error (*M* Exp. 1 = 21.0, *M* Exp. 2 = 19.5, Std. error = 1.54). It was concluded that participants in these groups were equivalent, which allows for comparisons between experiments.

Table 5. *Demographic Averages for Participants and Their Correlation with Problem Solving Performance in Experiment 2.*

| | Averages | | Correlation | |
|---|---|---|---|---|
| | *M* | *SD* | *r* | *p* |
| Gender | 64% male | - | .10 | .52 |
| Age | 20.0 | 2.3 | .02 | .88 |
| High School GPA | 3.83 | .21 | .14 | .39 |
| Year in College | 2.39 | 1.4 | -.07 | .65 |
| College GPA | 3.51 | .40 | .22 | .15 |
| Comfort with Computers (out of 7) | 4.27 | 1.6 | .01 | .97 |
| Expected Difficulty (out of 7) | 4.17 | 1.3 | .10 | .52 |
| Previous CS Courses | 80% taken 1 course | - | .12 | .42 |

The quality of subgoal labels was also consistent across the two experiments. In Experiment 1, participants in the guided constructive with hints conditions mainly created hint-term context-independent labels (45%) or context-independent labels (24%). Fewer of these participants created context-specific labels (22%) or incorrect labels (8%). In Experiment 2, all participants received the guided constructive with hints condition. Most of them created hint-term context-independent labels (33%) or context-independent labels (42%). Again, fewer participants created context-specific labels (22%) or incorrect labels (3%). These similarities suggest that the learners, regardless of the experiment in which they participated, were equivalent and that the instructions had the same effect on them.

**3.2.1 Learner-created labels improve problem solving performance.** The problem solving assessment in Experiment 2 was the same as in Experiment 1. It was scored using the same procedure, and the maximum possible score was 25. A main effect of guidance on practice problems was found, $F(2, 57) = 7.42$, $MSE = 23.8$, $p = .001$, partial $\eta^2 = .21$ (see Figure 11). Using the LSD post hoc procedure, participants who received scaffolding with their own constructed labels ($M = 23.7$, $SD = 3.66$) performed statistically better than those who received scaffolding with experimenter-created labels ($M = 17.9$, $SD = 5.78$; Mean Difference = 5.75, $p < .001$) or no scaffolding ($M = 19.5$, $SD = 4.98$; Mean Difference = 4.20, $p = .009$). Participants who received scaffolding with experimenter-created labels did not perform statistically differently from those with no scaffolding, Mean Difference = 1.55, $p = .32$.

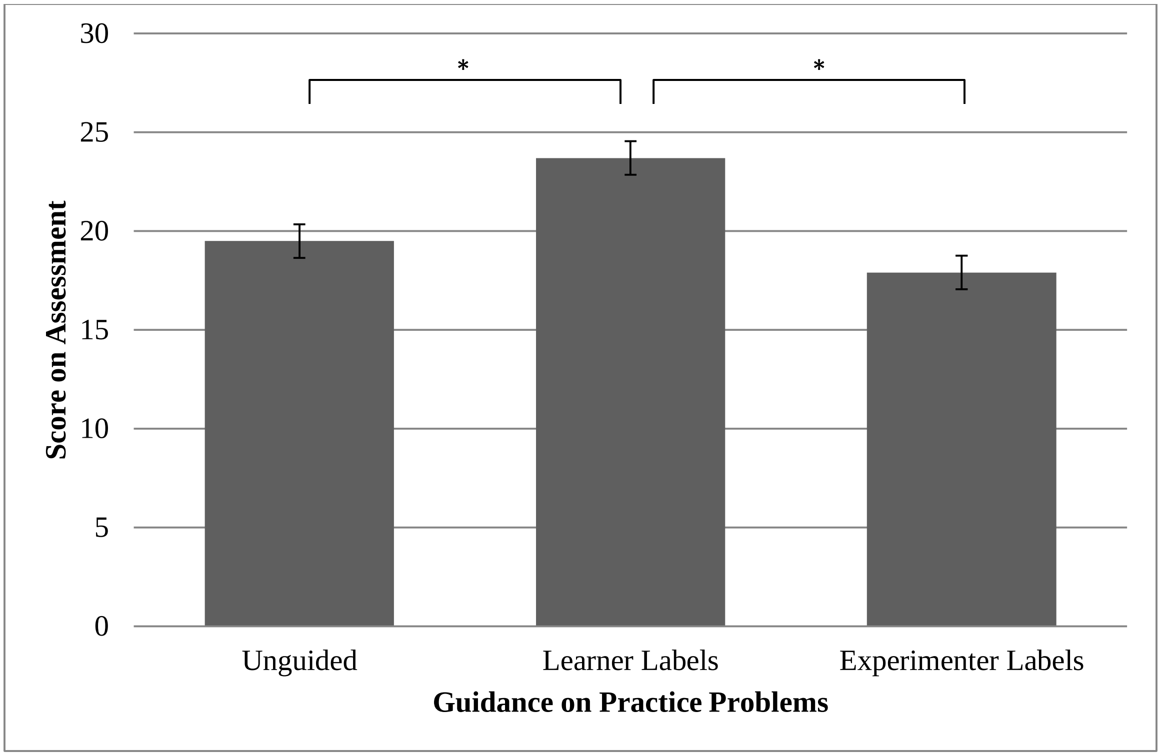


*Figure 11*. Performance on problem solving tasks among conditions in Experiment 2. Maximum possible score was 25. Error bars are standard error. Statistically significant differences are indicated with asterisks.

These findings suggest that when learner-created labels are used to scaffold initial problem solving, later problem solving improves. This improvement had a relatively large effect size too, meaning that the scaffolding accounts for a large portion of the differences among groups. For learners who constructed subgoal labels with the support of hints, one the best performing conditions from Experiment 1, problem solving performance can still be significantly improved by scaffolding initial problem solving with learner-created labels. Furthermore, the superior performance by participants who received their own subgoal labels as scaffolds provided further evidence that participants who constructed their own labels with hints created an effective mental organization of information related to the procedure. Their labels were conceptually-relevant enough that the labels could be applied as effective scaffolds to practice problems. Effective scaffolds help students to apply procedural knowledge to novel problems (Pea, 2004), and if participant-created labels can serve this purpose, then they must be high quality.

These results suggest that when learners create subgoal labels with enough support, such as the hints provided in this study, they should not be exposed to experimenter-created labels. Scaffolding with experimenter-created labels did not improve problem solving in this experiment over un-scaffolded practice problems, suggesting that the experimenter-created labels did not help guide participant problem solving. Imposing experimenter-created labels on participants who had developed their own effective organization of the procedure did not seem to help the participants in any way.

**3.2.2 No differences in time on task.** The time that participants spent completing each part of the procedure was recorded. Participants spent an average of 33.9 minutes (*SD* = 6.57) completing the subgoal training and using the worked example to construct subgoal labels. There was no main effect of guidance on practice problems during this period, $F(2, 57) = .84$, $MSE = 43.5$, $p = .44$, which is expected because there were no instructional differences among conditions at this point. During the period in which participants solved the practice problems, they took an average of 9.77 minutes (*SD* = 3.97). There was no main effect of guidance on practice problems, $F(2, 57) = .47$, $MSE = 16.1$, $p = .63$, meaning that the different conditions under which participants solved practice problems did not affect the amount of time it took them to complete the practice problems. During the problem solving assessment, participants took an average of 23.4

minutes (*SD* = 2.26). There was no main effect of guidance on practice problems for this time period, $F(2, 57) = 1.47$, $MSE = 5.01$, $p = .24$. Therefore, the participants who had practice problems scaffolded with their own labels did not take longer to complete the practice problems or the problem solving assessment, but they still performed better on the problem solving assessment than participants in other conditions.

**3.2.3 No differences in other metrics.** Similar to Experiment 1, no differences among conditions were found for the other metrics collected in the study. For the explanation assessment with a maximum possible score of 20, the mean score was 16.1 (*SD* = 4.02). There was no main effect of guidance during problem solving, $F(2, 57) = .32$, $MSE = 16.7$, $p = .73$, partial $\eta^2 = .01$. These results suggest that all participants were equally prepared to complete the explanation task. Because all participants received the same instructional materials except scaffolding on the practice problems, they were not expected to perform differently on this assessment.

No differences were found among group for participants' self-reported cognitive load, which was measured after the instructional period and before the problem solving assessment. Out of a possible rating of 100, the mean score was 38.9 (*SD* = 12.1), $F(2, 57) = .16$, $MSE = 151.2$, $p = .85$. In addition, no differences were found within each of the three types of cognitive load: intrinsic, extraneous, and germane (see Table 6). These results suggest that participants did not perceive differences in cognitive load among the conditions.

Table 6. *ANOVA Results for Intrinsic, Extraneous, and Germane Cognitive Load Measures for Experiment 2.*

| | *M* | *SD* | *F* | *p* |
|---|---|---|---|---|
| Intrinsic Load | 10.1 (out of 30) | 6.4 | .76 | .47 |
| Extraneous Load | 5.3 (out of 30) | 4.5 | .08 | .93 |
| Germane Load | 23.5 (out of 40) | 7.6 | .19 | .83 |

After the instructional period, participants rated how well they understood the instructions from “1 – Not well at all” to “7 – Very well.” Participants rated that they understood the instructions well ($M = 5.93$, SD = 1.1). These ratings were not affected by the guidance of practice problems, $F(2, 57) = .70$, $MSE = 1.14$, $p = .50$. Participants also rated how comfortable they were solving novel problems using the procedure from “1 – Not comfortable at all” to “7 – Very comfortable.” Participants rated that they were comfortable solving new problems ($M = 5.8$, SD = 1.1). These ratings were not affected by the guidance of practice problems, $F(2, 57) = .91$, $MSE = 1.19$, $p = .41$. These results indicate that, on average, participants in different conditions felt equally prepared to solve novel problems regardless of the guidance that they received when solving the practice problems.

# CHAPTER 4: CONCLUSIONS

Subgoal learning has been primarily supported through passive methods: subgoal labeled instructions. These methods have been successful at improving problem solving performance in procedural domains presumably because they give learners beneficial instructional guidance (e.g., Catrambone, 1998). Passive methods, however, are typically less effective than active and constructive learning methods (Chi, 2009). The primary goal of the present study was to further improve problem solving performance by exploring active and constructive methods of learning subgoals. The results suggest that guided constructive methods of learning subgoals can lead to better problem solving performance compared to passive, active, and unguided constructive methods. This finding means that learners can benefit from instruction that guides them to self-explain what instructions would typically directly explain. Guided constructive methods of learning subgoals were most effective when the instructions either provided hints while learners were creating labels or feedback after they created labels, but not when the instructions provided both. This finding supports the idea that providing too much instructional guidance can hinder learning.

The present experiment taught college-level, novice learners to program using Android App Inventor. The results, therefore, suggest that constructive learning can be better than passive learning, even for a complex problem solving procedure and even for novice learners. For a task that has a different level of complexity or for learners at a different level of knowledge, the results might have turned out differently. For example, if the task was more complex, learners might have needed more support to learn subgoal constructively, or constructive learning might be less effective in general than passive learning. In contrast, if the task was less complex, learners might not have needed as much support to understand the procedure well and might benefit from having less instructional support and more opportunities to construct knowledge. Similarly, for learners with more knowledge, providing less instructional support is typically associated with better learning because students have more opportunities to construct knowledge for themselves. An example of this is the expertise reversal effect (Sweller, 2010).

The pattern of results found in Experiment 1 adds to the argument that feedback can hinder learning. Although it is not uncommon to find that feedback hinders learning, usually the cause is attributed to learners' overreliance on feedback as a form of instructional support (e.g., Schworm & Renkl, 2006). In the present study, however, learners were not aware that they would receive feedback until after they finished studying the instructions, meaning that they could not rely on the information provided through feedback. Still the results show that when learners received hints during the constructive learning activity, receiving feedback hindered their later problem solving performance. This finding provides evidence that feedback can hinder learning when it provides unneeded additional instructional support.

Feedback might have hindered learning in this case because it required learners who had created good, context-independent self-explanations to compare their explanations with that of the experimenter. The reasons that this comparison could be detrimental were not explored in this research, but a likely explanation will be discussed here. Even if the explanations created by the participants and those created by the experimenter were similar, participants might not have had enough domain knowledge to recognize how similar the explanations were. For participant who created good, context-independent subgoal labels, comparing the two explanations could have had two effects: cause confusion in the learner who is unable to reconcile the explanations that they created and those that the experimenter created and/or cause the learner to abandon their explanations and use what they might have perceived to be the correct explanations. Both of these effects would negate the benefits of constructive learning – building knowledge upon prior knowledge in an organization that makes sense to the learner.

To explore whether the comparison between good participant-created and experimenter-created explanations is the cause of feedback's negative effect when learners received the guided constructive example with hints, a yolked experimental design could be employed. In this design, participants could be given the guided constructive with hints condition and asked to create their own subgoal labels. Then participants would be grouped into yolked pairs and receive either feedback based on the labels that they had created or the same feedback that their yolked partner had received. The feedback based on participant-created labels would be advertised as correct labels developed by an expert, but the feedback labels would be based on the labels created by the participant. Assuming that the labels created by participants were correct, minimal lexical changes could be made to the labels to make them look nominally different but still be conceptually similar. For example, if a participant created a label "add properties to the app," the feedback label might be "add properties of app."

It is hypothesized that participants who received feedback based on their labels would not have difficulty integrating their created labels with the feedback labels; therefore, it is hypothesized that this group would perform similarly to those who did not receive feedback. If some participants in this group created context-specific or incorrect labels, the feedback would default to the experimenter-created labels. In that case, these participants might perform better on novel problem solving tasks because their context-specific or incorrect labels are corrected by experimenter-created labels. It seems unlikely that providing feedback for learners who created good subgoal labels would further improve problem solving performance unless the learners were uncertain of their labels and could benefit from validation of their labels.

For participants who receive yolked feedback, it is hypothesized that they would perform as poorly or worse than participants in Experiment 1 who received experimenter-created labels as feedback. These participants would receive feedback labels that would be different enough from their own that it is expected that the participants would have trouble reconciling the two sets of labels. These participants might even perform worse because the feedback labels would be created by another participant who is a novice in the subject matter and in making subgoal labels; therefore, it is possible that the yolked feedback labels would make even less sense than experimenter-created labels to the participants in the yolked condition.

When feedback was not paired with hints in the guided constructive conditions, though, it improved problem solving performance. The study found no differences in problem solving performance between learners who received hints during the constructive learning activity or those who received feedback after the constructive learning activity. Therefore, there is no evidence that one type of instructional support is better than the other for learning. The quality of subgoal labels created by participants, however, was better when learners received hints than when they did not. This difference in subgoal label quality was not related to performance on any of the metrics in the present study, but it does suggest that participants had a better mental organization of information related to the procedure. It is tenable that future work could find that higher quality labels are related to better retention or performance on related problem solving procedures. It is also tenable that the feedback improved learners' mental organizations and no meaningful differences among the learners persisted after the feedback was given.

For learners who received hints while constructing their own subgoal labels, Experiment 2 found that participant-created labels could be used to effectively scaffold initial problem solving while learners solved practice problems. For these learners, experimenter-created labels did not effectively scaffold initial problem solving. This finding is similar to findings from the memory literature on subjective organization. In the subjective organization research, people are better able to recall words when they use their own method of organizing the words than when they are told to use a prescribed organization (e.g., if they were told to recall the words in alphabetical order; Tulving, 1962). In the subjective organization literature, the words being memorized are unrelated, meaning that there is not a correct way to organize them. In the present study, however, there is a correct conceptual understanding of the procedure, making the similarity in results interesting because it suggests that the learner's organization can guide the learner better than an expert's organization, even when there are incorrect ways or organizing information.

The results suggest that learners who receive hints for how to organize knowledge about the procedure perform better on later problem solving when they practice solving problems scaffolded with their own organization than when they practice solving problems scaffolded with an organization created by an instructional designer. This finding implies that learners' organizations of knowledge are as effective as the organization provided by the instructional designer or that learners' organizations are better for guiding problem solving even if they are inferior to the instructional designer's organization. This outcome is similar to the result in Experiment 1 in which the experimenter-created labels hindered problem solving performance when they were used as feedback for participant-created labels. In both cases, the organization from an instructional designer hurt problem solving performance.

Future work should focus on exploring whether constructive methods of subgoal learning could be developed into a general learning strategy. Perhaps teaching learners to create their own subgoal labels would help them to improve performance on a range of tasks. After constructing, with adequate support, subgoal labels for several procedures, learners might become skilled at developing labels and would be able to construct labels for new procedures in different domains without help from instructors or instructional designers. This strategy would likely have benefits that are similar to training students to self-explain in procedural domains. Training learners on this type of learning strategy could help learners perform better--both at initial learning and later transfer--across a range of procedural fields.

The present study suggests that learners are better able to solve novel problems when they learn the subgoals of a procedure through constructive methods that provide an optimal level of instructional support as guidance than when they learn subgoals through passive or active methods. Providing hints to learners that help them to realize the similarities between different instances of subgoals would be an easy intervention to include in instructional material because it does not need to be customized for each individual learner. If providing hints helps students learn constructively as much as providing feedback, as this study suggests, then constructive learning can be supported in a larger range of learning environments. Much of the constructive learning research has been done in face-to-face learning environments in which the instructor can scaffold students to construct knowledge. By providing hints to learners, students can constructively learn subgoals in learning environments that do not provide feedback, or at least not immediate feedback, like many online learning environments. If future work suggests that constructing subgoal labels can be a general learning strategy applied in various domains, then the learning methods in the present research will become even more compelling. Based on the findings of the present study, the best subgoal learning outcomes should be achieved through constructive methods with some, but not too much, instructional support.

## APPENDIX A

*Training to Create Subgoal Labels*

### How to Make Subgoal Labels

Research has shown that when studying problem solving procedures, like creating apps in Android App Inventor, the best learning happens when students explain to themselves (self-explain) the purpose of steps in the procedure. Successful self-explanations identify the subgoals of the procedure. Subgoals are components of the problem solution (the overall goal) that are made up of individual steps taken to solve the problem (such as adding two numbers together). That means individual steps make up subgoals, and subgoals make up the solution.

For instance, if you were asked to solve for x in the equation, $2x + 4 = 6x + 10$, you would use the following steps

$$\left.\begin{array}{l} 6x + 10 = 2x + 4 \\ \quad -10 \qquad -10 \\ \quad -2x \qquad -2x \\ 6x - 2x = 4 - 10 \end{array}\right\} \text{Isolate variable}$$

$$\left.\begin{array}{l} 4x = -6 \\ /4 = \ /4 \\ x = -3/2 \end{array}\right\} \text{Simplify terms}$$

Each group of steps is a subgoal of the problem. The labels in this example ("Get variables on same side," "Simplify," and "Get variable with coefficient of 1") describe the purpose of the subgoals. A good subgoal label describes the function or the goal of each group of steps. The label should convey what the steps achieve toward solving the problem to help the learner connect steps of the procedure to their purpose.

While you are learning to create apps, you will be asked to provide your own subgoal labels for the examples that you receive. To do this, you will be asked to identify the purpose of groups of steps in the examples (label the subgoals). Good subgoal labels are action-based phrases (i.e., similarly to imperative sentences like "Close the door," or "Press the button"); they tell the problem solver what to do next. The following activity is intended to give you practice in making your own subgoal labels.

**Activity**

Label the groups of steps in the following example using the same subgoal labels from the previous example.

Solve for x

$$\left.\begin{array}{l} 4x - 8 = 2x + 6 \\ \quad +8 \qquad +8 \\ \quad -2x \qquad -2x \\ 4x - 2x = 6 + 8 \end{array}\right\}$$

$$\left.\begin{array}{l} 2x = 14 \\ /2 = \ /2 \\ x = 7 \end{array}\right\}$$

ANSWER
Solve for x

$$\left.\begin{array}{l} 4x - 8 = 2x + 6 \\ +8 \qquad\quad +8 \\ -2x \qquad\quad -2x \\ 4x - 2x = 6 + 8 \end{array}\right\} \text{Isolate variable}$$

$$\left.\begin{array}{l} 2x = 14 \\ /2 = \ /2 \\ x = 7 \end{array}\right\} \text{Simplify terms}$$

For this order of operations problem, create subgoal labels for each group of steps (by labeling each group of steps with its purpose).
Solve for x

x = 4 * (5-2) + 12 / (4-1) – 7

(5-2)

(4-1)

x = 4 * 3 + 12 / 3 – 7

4 * 3

12 / 3

x = 12 + 4 – 7

x = 11

ANSWER

Solve for x

x = 4 * (5-2) + 12 / (4-1) – 7

(5-2) — Simplify parentheses (and exponents) operations

(4-1)

x = 4 * 3 + 12 / 3 – 7

4 * 3 — Simplify multiplication and division operations

12 / 3

x = 12 + 4 – 7 — Simplify addition and subtraction operations

x = 11

Now that you have some practice applying and creating subgoal labels, it's time to make subgoal labels for creating apps. The examples that you will be given all have the same subgoals, but this doesn't mean that you have to stick to the subgoal labels that you create on the first example. Please feel free to update your subgoal labels as you learn more.

## APPENDIX B

*Training to Complete Verbal Analogies*

### Verbal Analogies

Verbal analogies provide excellent training in seeing relationships between concepts. Verbal analogies were previously used to test cognitive ability on standardized tests (like the SAT, the GRE, and other professional exams). Increasingly, too, employers may use these word comparisons on personnel and screening tests to determine an applicant's quickness and verbal acuity.

<u>How to "Read" Analogies</u>
The symbol ( : ) means "is to" and the symbol ( : : ) means "as." Thus, the analogy, "aspirin : headache : : nap : fatigue," should be read "aspirin is to headache as nap is to fatigue." Stated another way, the relationship between aspirin and headache is the same as the relationship between nap and fatigue.

<u>Tips for Doing Analogies</u>

- Try to determine the relationship between the complete pair of words.
- Eliminate any pairs in your answer choices that don't have the same relationship.
- Try putting the pairs into the same sentence: "Aspirin relieves a headache." Therefore, a nap relieves fatigue.
- Sometimes paying attention to the words' parts of speech helps. For example, "knife" (noun) : "cut" (verb) : : "pen" (also a noun) : "write" (also a verb).

<u>Common Relationships Between Word Pairs</u>

| Relationship | Example |
|---|---|
| Sameness (synonyms) | wealthy : affluent : : indigent : poverty-stricken |
| Oppositeness (antonyms) | zenith : nadir : : pinnacle : valley |
| Classification Order (general - specific) | orange : fruit : : beet : vegetable |
| Difference of Degree | clever : crafty : : modest : prim |
| Person Related to Tool, Major Trait, or Skill or Interest | entomologist : insects : : philosopher : ideas |
| Part and Whole | eraser : pencil : : tooth : comb |
| Steps in a Process | cooking : serving : : word processing : printing |
| Cause and Effect (or Typical Result) | fire : scorch : : blizzard : freeze |
| Thing and Its Function | scissors : cut : : pen : write |
| Qualities or Characteristics | aluminum : lightweight : : thread : fragile |
| Substance Related to End Product | silk : scarf : : wool : sweater |

| | |
|---|---|
| Implied Relationships | clouds : sun : : hypocrisy : truth |
| Thing and What It Lacks | atheist : belief : : indigent : money |
| Symbol and What It Represents | dove : peace : : four-leaf clover : luck |

**Activity**

1. happiness : smile : : ________________ : frown
   - worry
   - terror
   - mood
   - temper
   - encomium

2. water : ________________ : : food : hunger
   - element
   - drink
   - starvation
   - liquid
   - thirst

3. government : ________________ : : media : news
   - rule
   - bureaus
   - people
   - laws
   - legislature

4. light bulb : electricity : : car : ________________
   - oil
   - motor
   - wheels
   - generator
   - gasoline

5. chapter : book : : ________________ : nation
   - state
   - country
   - kingdom
   - president

6. tall : short : : _________________ : smooth

- deep
- rough
- texture
- wide

7. preserve : waste : : ordinary : _________________

- special
- recycle
- expensive
- usual

8. _________________ : freeze : : horrible : wonderful

- stop
- cold
- terrible
- boil

9. _________________ : ballerina : : soft : velvet

- graceful
- dancer
- performance
- edgy

10. genius : _________________ : : glass : clear

- intelligent
- brains
- capable
- slow

## REFERENCES


Aleven, V. A. W. M. M., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*, 147-179.

Atkins, S. M., Sprenger, A. M., Colflesh, G. J. H., Briner, T. L., Buchanan, J. B., Chavis, S. E., … Dougherty, M. R. (2014). Measuring working memory is all fun and games: A four-dimensional spatial game predicts cognitive task performance. *Experimental Psychologist, 61*(6), 417-438.

Atkinson, R. K. (2001). Optimizing learning from examples: Using animated pedagogical agents. *Journal of Educational Psychology*.

Atkinson, R. K., Catrambone, R., & Merrill, M. M. (2003). Aiding transfer in statistics: Examining the use of conceptually oriented equations and elaborations during subgoal learning. *Journal of Educational Psychology, 95*(4), 762-773.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of the Educational Research, 70*(2), 181-214. doi:10.2307/1170661

Berthold, K., Eysink, T. H. S., & Renkl, A. (2009). Assisting self-explanation prompts are more effective than open prompts when learning with multiple representations. *Instructional Science, 37,* 345-363. doi:10.1007/s11251-008-9051-z

Bielaczyc, K., Pirolli, P. L., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. *Cognition and Instruction, 13*(2), 221-252. doi:10.1207/s1532690xci1302_3

Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. J. Metcalfe & A. P. Shimamura (Eds.). *Metacognition: Knowing about Knowing*. MIT Press: Cambridge, MA.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.) (2000). *How people learn: Brain, mind, experience, and school: Expanded edition.* Retrieved from http://www.nap. edu/catalog.php?record_id=9853

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory and Cognition, 22*, 605-615. doi:10.3758/BF03198399

Catrambone, R. (1995). Aiding subgoal learning: Effects on transfer. *Journal of Educational Psychology, 87*(1), 5-17. doi:10.1037/0022-0663.87.1.5

Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22,* 1020-1031. doi:10.1037/0278-7393.22.4.1020

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General, 127,* 355-376. doi:10.1037/0096-3445.127.4.355

Catrambone, R., Gane, B. D., Adams, A. E., Bujak, K. R., Kline, K. A., & Eiriksdottir, E. (2016). Task analysis by problem solving (TAPS): A method for uncovering expert knowledge. *Under review*.

Catrambone, R. & Holyoak, K. J. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*(6), 1147-1156.

Chi. M. T. H. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science, 1*(1), 73-105.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*, 145-182.

Chi, M. T. H., de Leeuw, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439-477.

Conati, C., & VanLehn, K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education, 11,* 389-415.

de Koning, B. B., Tabbers, H. K., Rikers, R. M. J. P., & Paas, F. (2011). Improved effectiveness of cueing by self-explanations when learning from a complex animation. *Applied Cognitive Psychology, 25*, 183-194.

deWinstanley, P. A., Bjork, E. L., & Bjork, R. A. (1996). Generation effects and the lack thereof: The role of transfer-appropriate processing. *Memory, 4,* 31-48.

Durso, F. T., Rea, C. B., & Dayton, T. (1994). Graph-theoretic confirmation of restructuring during insight. *Psychological Science, 5*(2), 94-98.

Eiriksdottir, E., & Catrambone, R. (2011). Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors, 53*(6), 749-770. doi:10.1177/0018720811419154

Hausmann, R. G. M., & Chi, M. T. H. (2002). Can a computer interface support self-explaining? *Cognitive Technology, 7*(1), 4-14.

Hundhausen, C. D., Farley, S. F., & Brown, J. L. (2009). Can direct manipulation lower the barriers to computer programming and promote transfer of training?: An experimental study. *ACM Transactions in CHI, 16*(3). doi:10.1145/1592440.1592442

Jacoby, L. L. (1978). On interpreting the effects of repetition: Solving a problem versus remembering a solution. *Journal of Verbal Learning & Verbal Behavior, 17*, 649-667.

Kirschner, P., Sweller, J., & Clark, R. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, *41*(2), 75-86.

LeFevre, J., & Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction, 3*, l-30. doi:10.1207/s1532690xci0301_1

Linn, M., & Clancy, M. (1992). The case for case studies of programming problems. *Communications of the ACM*, *35*(3), pp 121-132.

Margulieux, L. E. & Catrambone, R., (2014). Improving programming instruction with subgoal labeled instructional text. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.) *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (pp. 952-957). Austin, TX: Cognitive Science Society.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (pp. 71-78). New York, NY: Association for Computing Machinery. doi:10.1145/2361276.2361291

Margulieux, L. E., McCracken, W. M., & Catrambone, R. (2015). Mixing in-class and online learning: Content meta-analysis of outcomes for hybrid, blended, and flipped courses. In *Proceedings of Computer Supported Collaborative Learning*. International Society of the Learning Sciences.

Maxwell, S. E., & Delaney, H. D. (2004). *Designing experiments and analyzing data: A model comparison perspective* (2nd ed.). New York, NY: Psychology Press.

Morrison, B. B., Dorn, B., & Guzdial, M. (2014). Measuring cognitive load in introductory CS: adaptation of an instrument. In *Proceedings of the Tenth Annual Conference on International Computing Education Research*, 131–138.

Palmiter, S., Elkerton, J., & Baggett, P. (1991). Animated demonstrations versus written instructions for learning procedural tasks: A preliminary investigation. *International Journal of Man-Machine Studies, 34,* 687-701. doi:10.1016/0020-7373(91)90019-4

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences, 13*(3), 423-451. doi: 10.1207/s15327809jls1303_6

Pirolli, P., & Recker, M. (1994). Learning strategies and transfer in the domain of programming. *Cognition and Instruction, 12*(3), 235-275.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science, 21*(1), 1-29.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction, 12*, 529-556.

Renkl, A. (2005). The worked-out-example principle in multimedia learning. In R. Mayer (Ed.), Cambridge handbook of multimedia learning. Cambridge, UK: Cambridge University Press.

Renkl, A., & Atkinson, R. K. (2002). Learning from examples: Fostering self-explanations in computer-based learning environments. *Interactive Learning Environments, 10*(2), 105-199. doi:10.1076/ilee.10.2.105.7441

Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skills acquisition: A cognitive load perspective. *Educational Psychologist, 38*, 15–22.

Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: The effects of example variability and elicited self-explanations. *Contemporary Educational Psychology, 23*, 90-108.

Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting factors that predict success and failure in a CSI course. *SIGCSE Bulletin, 33*(4), pp 101-104.

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: When instructional explanations reduce self-explanations. *Computers & Education, 46*, 426-445.

Soloway, E. (1986). Learning to program = learning to construct mechanisms and explanations. *Communications of the ACM, 29*(9), 850-858.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review, 22*(2), 123-138.

Tew, A. E., & Guzdial, M. (2011). The FCS1: A language independent assessment of CS1 knowledge. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (SIGCSE '11). ACM, New York, NY, USA, 111-116. doi:10.1145/1953163.1953200

Tulving, E. (1962). Subjective organization in free recall of “unrelated” words. *Psychological Review, 69*(4), 344-354.

Wylie, R., & Chi, M. T. H. (2014). The self-explanation principle in multimedia learning. In R. Mayer (Ed.) *The Cambridge Handbook of Multimedia Learning, 2nd Edition (pp.413-432).* Cambridge University Press.

Yeh, Y. F., Chen, M. C., Hung, P. H., & Hwang, G.-J. (2010). Optimal self-explanation prompt design in dynamic multi-representational learning environments. *Computers & Education*, *54*(4), 1089–1100.