JTH

https://jthkkss.com/
ISSN 2710-6357

Exploring Two Worked Example Designs for Learning Introductory Programming from Students' Perspectives

Nainan, Mariam[1*], Balakrishnan, Balamuralithara[1], & Mohamad Ali, Ahmad Zamzuri[1]

[1]Faculty of Art, Computing & Creative Industry, Sultan Idris Education University, Tanjong Malim, 35900, MALAYSIA
*Corresponding author email: marnai2004@yahoo.com

Received 29 October 2020; Accepted 10 November 2020; Available online 10 November 2020

Abstract: Worked examples are effective for learning problem solving but, only if students engage with the content. An approach to promote engagement is through signalling. This study compared worked example designs for learning introductory programming using two approaches for signaling: labelled and visualised. It explored students' preferences and perceptions of the designs through a crossover design where students were exposed to both worked example designs. Data was collected through a questionnaire. Quantitative analysis showed that more students favoured visualised design. Qualitative analysis showed that students found both designs helped to understand the solution. Additionally, visualised worked examples also helped in understanding the problem, the relationship between problem and solution, as well as the programming process. Other differences were also identified.

Keywords: Worked examples, signalling, programming education

1. Introduction

Research has shown that students who study worked examples have benefited more in relation to learning how to solve problems than students who attempted to solve the problems on their own (Renkl, 2014). A worked example contains not only a demonstration of domain concepts, principles, or procedures, but also a specific problem for which these are applied. In the context of the programming domain, the demonstration would contain a complete program that solves the given problem (Skudder & Luxton-Reilly, 2014). The program would show the application of different programming concepts and their implementation in a programming language so that students are given a demonstration of how the concepts are applied and implemented in the language.

Since the worked example already has the solution, students do not need to make the effort to solve the problem themselves, unlike the case of learning through problem solving. However, students do need to invest cognitive resources to process the information given in the worked example in order to gain a proper understanding of how the solution has solved the problem. They need to recognise the concepts demonstrated in the solution. They must explain to themselves how these concepts have been applied so that they remember and learn how and for what type of problems they are applicable. This process of self-explanation is the foundational premise for the effectiveness of learning from worked examples, or example-based learning (Renkl, 2014). In other words, students should not simply read the worked example content. They must be cognitively engaged with the content.

Worked examples must be designed in such a way as to encourage students to self-explain. Subgoal labelled worked example design (Atkinson, Catrambone, & Merrill, 2003; Catrambone 1998) has been found to be helpful to encourage students to explain to themselves how the solution solves the problem. In a labelled worked example, groups of related statements are clearly demarcated and labels are inserted to explain the purpose of that group (Morrison, Margulieux, & Guzdial, 2015). The purpose would be stated as a subgoal. Hence, the label is called a subgoal label. A subgoal represents a mini-problem or subproblem. In other words, it is small aspect of the overall problem. The group of statements represents a subsolution that achieves the subgoal. The intention of inserting subgoal labels is to assist

*Corresponding author: marnai2004@yahoo.com

students in understanding that the solution is made up of subsolutions and that the various subsolutions achieve different subgoals, which collectively contribute to achieving the overall goal.

Subgoal labels may be regarded as signals (Margulieux & Catrambone, 2016) for drawing students' attention to the group of program statements. The use of signals, or signalling, is an instructional technique for alerting students to important information in instructional text as well making the structural organisation of the information more obvious to students (Lorch, Lemarié, & Grant, 2011; Schneider, Beege, Nebel, & Rey, 2018). The subgoal labels in the solution program should draw the students' attention to the group of statements and assist them to understand those statements as well as how they relate to the rest of the program.

The current study proposed visualised worked example design as another approach that uses signals to draw students' attention to relevant information in worked example content in order to encourage students to cognitively process the information. A visualised worked example would contain a problem section and solution section, containing a complete program, just as in a labelled worked example. However, no labels would be inserted in the program. Instead, there will be an additional problem analysis section, where the subgoals or subproblems will be listed, in order to demonstrate to students how to break a problem down into subproblems. The purpose of removing the labels from the program into a separate problem analysis section is to demonstrate and place emphasis on problem analysis. The intention is to address the issue of a lack of problem analysis skill among students learning programming. Students who do not spend enough time analysing a problem may expend unproductive time attempting to find a solution because they do not fully understand the problem (Loksa & Ko, 2016). Students need to understand the requirements of the problem before attempting the solution (Hanks & Brandt, 2009). The lack of problem analysis skill is also found in other domains where problem solving is an integral component of teaching and learning. For example, in chemistry education, Yuriev, Naidu, Schembri, and Short (2017) reasoned that students may end up with incorrect solutions because they ignore certain aspects of the problem due to inadequate problem analysis.

In visualised worked example design, in order to link the subproblem to the statements in the program that address that subproblem, signalling in the form of text highlighting is used. And, this highlighting is interactive. These features are achieved through the use of web technology. The relevant program statements are highlighted when the student selects a subproblem in the analysis section. More specifically, whenever the student moves the mouse over a subproblem, the program statements that are linked to that subproblem are highlighted. So, signalling is interactive in the sense that signals are activated in response to the student's actions. On the other hand, in labelled worked examples, the signals are fixed or static as labels. In visualised worked examples, the signals are activated, deactivated, or reactivated, in accordance to the student's needs. It was expected that the interactive signals would encourage students to engage with the learning content.

The current study sought to compare labelled and visualised worked example designs by exploring students' perspectives of the two designs. Understanding students' perceptions of the worked example design to support learning may help to identify characteristics they perceived as useful (Liaw, 2008; Peart Rumbold, Keane, & Allin, 2017). It may shed light on the aspects of worked example study that students felt helped them in their learning though a comparison of the two worked example designs. Specifically, the current study sought to explore: (1) which worked example design (labelled or visualised) that more students selected for different aspects of learning, and (2) the reasons for their selections. The reasons given by students may give deeper insight into the learning experiences of the students. The empirical study employed a crossover design so that participants could experience both worked example designs and select between them. Students' sections and reasons were collected through questionnaires. The context of the study was teaching and learning introductory programming, which is the first programming course taken by students in undergraduate programmes leading to careers in computing-related fields. Introductory programming courses are now commonly mandatory for engineering and science undergraduate programmes as well. Hence, understanding how worked examples should be designed to support successful learning of introductory programming is essential.

2. Literature Review

2.1 Need to Emphasise Problem Analysis in Programming Process

The programming process is basically a problem solving process. Students in introductory programming courses need to be aware of the programming process, which involves problem analysis and solution generation. Medeiros, Ramalho, and Falcão (2019) identified problem solving as an important skill. They identified that learning how to analyse a problem and generate a solution may be challenging to students in an introductory programming course.

In their study of self-regulation behaviour of students during the programming process, Loksa and Ko (2016) identified that very few students performed the initial step of interpreting and clarifying the problem. In other words, those students did not properly analyse the problem to understand the requirements. Loksa and Ko (2016) also mentioned that this caused delays later when they were creating the solution because the students realised that they did not fully understand the problem. They suggested that further research is needed to consider whether teaching this step of analysing the problem leads to better problem solving. Furthermore, Denny, Becker, Craig, Wilson, and Banaszkiewicz (2019) called for research in teaching students how to break problems into subproblems and also why students have difficulties doing so. Similarly, Selby (2015) mentioned that problem decomposition is perceived as challenging possibly because of inexperience or inadequate understanding. Hence, research into finding a way to demonstrate to students how a problem may be broken down into subproblems (or problem analysis) is necessary. The current study on visualised worked example design sought to address this issue.

2.2 Signalling

Signals are instructional devices used in learning content to assist students as they read and process the content by emphasising what is important and making the organisational structure of the content more explicit (Lemarié, Lorch Jr, Eyrolle, & Virbel, 2008; Lorch, Lemarié, & Grant, 2011; Schneider, Beege, Nebel, & Rey, 2018). Signalling helps students distinguish between relevant and irrelevant information, comprehend information that is unfamiliar or new, and create mental representation of the information (Lemarié et al., 2008). Signals may emphasise, label, or identify sections of text, or show their relationships. However, student factors, such as background knowledge, learning goals, and motivation, may have an impact on the effectiveness of signals in learning content (Lemarié et al., 2008).

Signals may appear in text, pictures, or video (Schneider et al., 2018). Text-based and picture-based signals, mostly used in printed and multimedia, are typically static. Signals in video-based learning content are usually dynamic since they appear at different points in the video presentation, as and when needed. But, although the presentation is dynamic, the appearance of a video-based signal is fixed in relation its presentation sequence. In other words, whenever the video is replayed, the signal will appear in the same place and at the same time, with all its defined changes, as when played the first time. Visualised worked examples are designed to employ interactive signals. The signals are designed to appear interactively, that is, in response to students' actions.

The effectiveness of signalling for learning has been studied through experimental studies (e.g., review in Schneider et al. (2018)). Eye-tracking studies have also been used to explain how signals help draw attention and reduce effort in searching for relevant information (e.g., Ozcelik, Arslan-Ari, & Cagiltay, 2010). The current study sought empirical evidence from students' perspectives.

2.3 Subgoal Labelled Worked Examples for Teaching and Learning Programming

The use of subgoal labelled worked examples has recently been researched in the context of teaching and learning programming. Margulieux, Catrambone, and Guzdial (2016) studied the use of subgoal labels in worked examples for teaching a block-based programming language. In their experiment, the experimental group (using labelled version) outperformed the control group (using non-labelled version).

Morrison and colleagues conducted studies of worked examples using text-based programming languages. Morrison, Margulieux, and Guzdial (2015) explained that segmenting the solution program and inserting labels for each segment helped students to process the worked example content for more effective learning. The labels acted as signals to facilitate processing (Margulieux & Catrambone, 2016). In one study (Morrison, Margulieux, & Guzdial, 2015), the researchers divided students into three groups: those who studied worked examples with no labels, with labels given, and where labels were generated by students. The worked examples were interleaved with practice problems. Each of the three groups were split into two subgroups: isomorphic or transfer practice problems. The results of their study showed among the best performing groups were students who were given labels and transfer practice problems. In a similar study, Morrison, Margulieux, Ericson, and Guzdial (2016) found that the students given labels, and either type of practice problem, performed the best. Both studies were conducted among students in an introductory programming course. The current study also focussed on worked examples using a text-based programming language in an introductory programming course but sought to identify students' perceptions of labelled and visualised worked example designs.

3. Research Method

3.1 Research Design

An empirical study was conducted to explore two worked example designs (labelled or visualised) for learning introductory programming from the perspectives of students. In order for students to compare the two designs and state their selections and reasons for their selections, the study employed a crossover design similar to the design used for other education research studies. For example, Mathieson (2012) conducted a study where participants were given two types of feedback on assignments. The participants were divided into two groups. For the first section of the study, the first group was given one type of feedback and the second group was given the other type. During the second section of the study, the type of feedback given to the two groups was switched. In this way, both groups had received both types of feedback. At the end of the study, participants were given a questionnaire to indicate their preferred type of feedback. They were also asked to give reasons for their selections. Similarly, in a study conducted by Smith, Cavanaugh, and Moore (2011), the researchers investigated two different instructional delivery forms. Two groups of participants were first given one delivery form and then switched to second form. Participants were then given a questionnaire where they evaluated both types of instructional delivery forms. Prunuske, Henn, Brearley, and Prunuske, (2016) also studied two different instructional delivery methods in a crossover design. Participants were assigned to four different groups. Each group was given both types of delivery methods over four sessions but in different combinations. For the fifth session, participants were given both methods and allowed to select either one. At the end of the study, questionnaires were given to participants to determine their preferences and comments.

In a similar manner, in the current study, a crossover design was used. Participants were divided into two groups, named Group A and Group B. Both groups were given three worked examples to study. The worked examples were presented using the two different designs: labelled and visualised. For Group A, the first worked example was presented using visualised design whereas for Group B, it was presented using labelled design. For the second worked example, the design was switched. In other words, Group A studied a labelled worked example and Group B studied a visualised worked example. The purpose of switching was to control for the order of presentation of the different designs (Johnson & Christensen, 2014). In this manner, participants were exposed to both designs for the worked examples. For the third worked example, similar to the study in Prunuske et al. (2016), participants could select to view the worked example in either visualised or labelled design. It is noted that for all three worked examples, the problems and solutions were exactly the same. The only difference was in the design. This crossover design is illustrated in Fig. 1. Similar to the mentioned studies above, data was collected through a questionnaire.

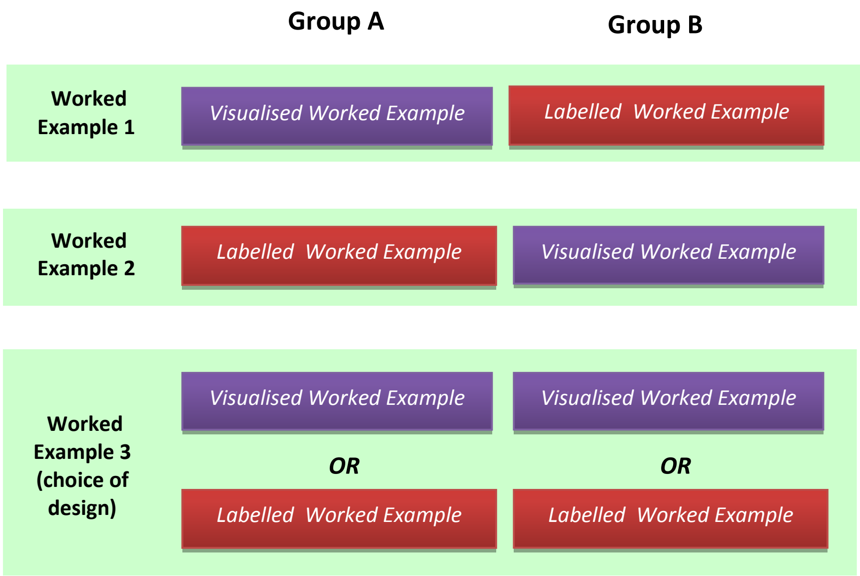

Fig. 1: Types of design used for worked examples presented to the two groups.

3.2 Participants

Participants were recruited from a university in Malaysia in the first semester of 2018 among undergraduate students enrolled in an introductory programming course. The course assumed that none of the students had any prior programming knowledge. In their weekly course schedule, students attended lectures and computer laboratory classes for hands-on programming sessions. In these sessions, students were given a review of the lecture topics covered in the previous week, after which they were expected to solve problems given in worksheets. The course covered fundamental programming topics, such as, selection and repetition control structures, and used a text-based programming language., The course was mandatory for all students and it was a prerequisite for other mandatory courses in their undergraduate programmes. The participants were from two intact classes in the course. One of them was designated as Group A and the other as Group B. Both of the classes were taught by the same lecturer. All participants gave informed consent for participation in the study and the study was given ethical clearance by the university's ethics review committee.

3.3 Learning Materials

The empirical study was conducted during one of the programming sessions. Each participant was presented with three worked examples based on three common patterns related to the "loop" concept. For the participants, the labelled and visualised designs were denoted as 3 sections presentation style and 4 sections with highlighting presentation style, respectively. A labelled worked example contained a problem, solution, and sample run sections, as shown in Fig. 2. The problem section contained the problem specification. The solution section contained the program. Labels were inserted as comments in the program to describe the purpose of one or more program statements that appeared below them. The sample run section contained a sample of the execution of the program in terms of input and output. (Input was indicated by underlining).

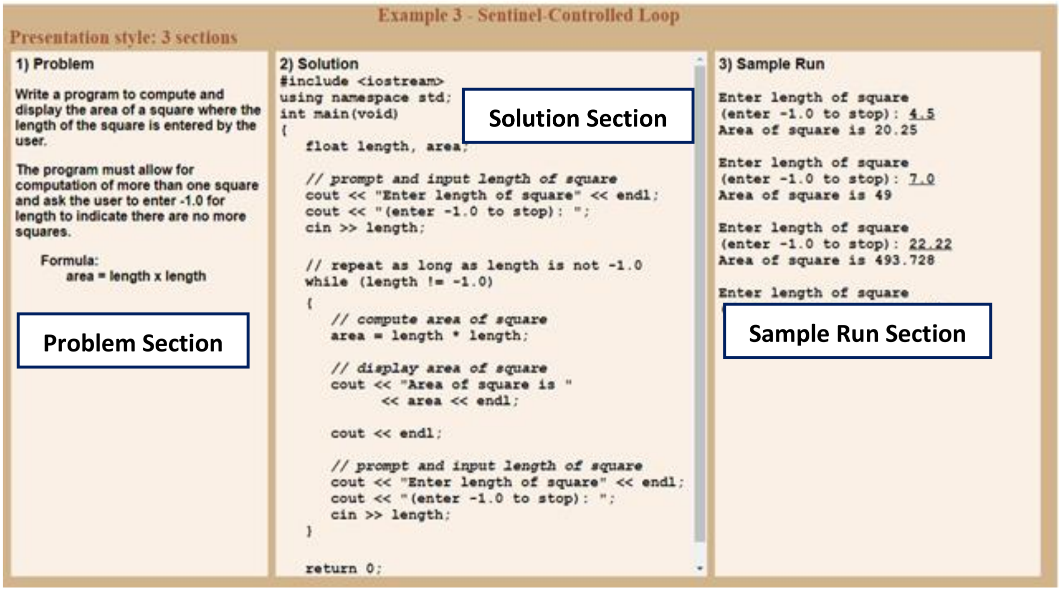

Fig. 2: Sample labelled worked example (or 3 sections style)

Similarly, a visualised worked example contained problem, solution, and sample run sections. The problem and sample run sections contained the exact same information as the labelled design version. However, the program in the solution section did not have labels. Instead, the visualised version had an additional analysis section. This section contained a list of subproblems that corresponded to the labels in the labelled design. A sample is shown in Fig. 3. The purpose of the analysis section was to demonstrate how problem analysis might be done. The subproblems were listed at two levels in a question and answer format. The question represented a subproblem category. The answer was the detail that was specific to that particular problem. By doing so, common subproblem categories, that are shared by problems of similar types, could be illustrated.

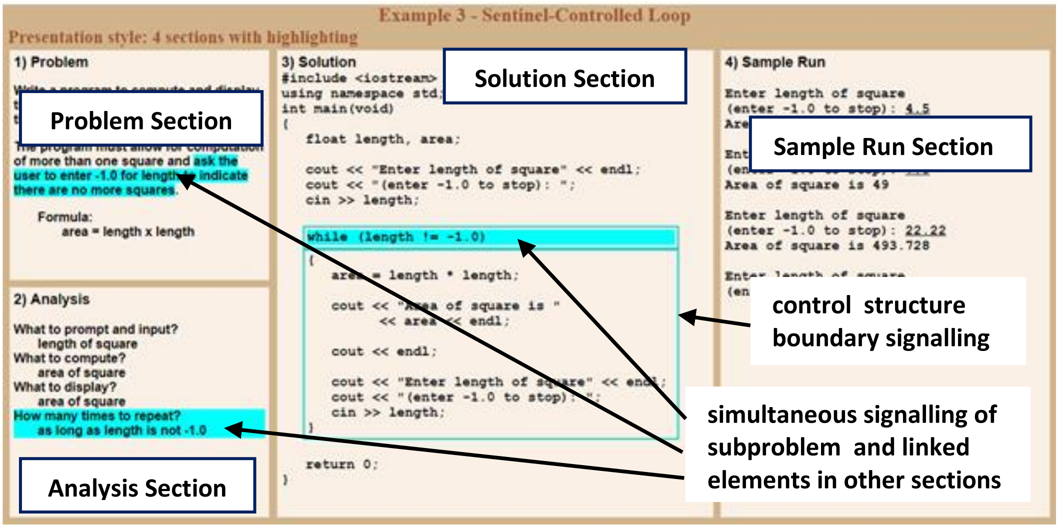

Fig. 3: Sample visualised worked example (or 4 sections with highlighting style)

Furthermore, visualised worked examples employed text highlights to link the subproblems to relevant elements in the other sections, as shown in Fig. 3. The highlights were activated when the student moved the mouse over a subproblem. Text highlighting was implemented for the subproblem question and answer, the statements in the program associated with that subproblem, and the elements in the problem specification relevant to the subproblem. In addition, the boundary of the loop control structure was highlighted as well. The purpose of boundary highlighting was to show to students the position of the statements in relation to the control structure. These highlights activated through interactivity are regarded as interactive signals.

Both labelled and visualised worked examples were implemented as web pages using HTML5 and CSS3. In addition, highlighting and interactivity in visualised worked examples were implemented using JavaScript. Highlights were implemented by turning the background colour of the relevant text to blue. It also entailed setting the outline of the control structure to a blue colour. Highlights were activated when stduents hovered the mouse over a subproblem in the analysis section. Students could select different subproblems to activate the relevant highlights for each of them.

3.4 Questionnaire

A questionnaire was used to collect data on participants' perspectives on the worked example designs. The questionnaire contained four questions. The first question asked participants which worked example design they preferred. The purpose was to obtain their overall impression. The second and third questions were related to perceived learning effectiveness and satisfaction, respectively. These aspects of learning from the participants' perspectives have been investigated in studies of technology-supported learning (e.g., (Liaw, 2008; Nugroho, Setyorini, & Novitasari, 2019)). The fourth question asked for their preference for possible future worked example study. All questions, except for the first, were adapted from the questionnaire used in (Mathieson, 2012). For the first three questions, participants were also asked to give reasons for their selections. Hence, the questionnaire collected both quantitative and qualitative data. The participants' selections for the questions represented the quantitative data. The reasons given by participants formed the qualitative data. All questions were reviewed by three experts for face and content validity.

3.5 Procedure

The study was conducted during a session, after the instructor had reviewed the "loop" concept. First, participants were introduced to the study, invited to participate, and completed demographic information forms. Next, they were asked to study the three worked examples. After the learning activity, participants completed a post-test and a questionnaire. Overall, the study lasted for 30 minutes. The post-test scores did not contribute to the grade for the course. This paper focuses only on the questionnaire responses and so the post-test and its results are not discussed further.

3.6 Data Analysis

Data collected from the questionnaires were analysed both quantitatively and qualitatively. Quantitative analysis involved computation of frequency distributions of participants' selections, by groups and in total. Qualitative data were analysed using qualitative content analysis (Schreier, 2012). After a first and second reading of the reasons given by participants for the first three questions, the first coder created a coding frame inductively from the data. The coding frame was then evaluated by a second coder. The second coder independently coded the whole data set and identified issues with the coding frame. These issues were discussed and the coding frame was revised where necessary. The first coder then recoded the complete data set with the revised coding frame. Next, a third coder coded the complete data set. Coding results of the first and third coders were compared, and wherever there were differences, they were discussed until consensus was reached.

4. Findings and Discussion

Questionnaires were collected from a total of 38 participants from the two groups, aged between 19 and 25 years, with 10 female and 28 male. Of these participants, 14 were from Group A and 24 from Group B.

4.1 Participants' Selection of Worked Example Design

The four questions asked participants to select between the two worked example designs, with regard to overall preference, perceived effectiveness for learning, satisfaction, and preference for future worked example study. Frequency counts and percentages of the selections made for each question for each worked example design, in total and by groups, are shown in Table 1. For overall preference, the results showed that a large majority (64% and above) preferred visualised worked example design in both Group A and Group B. Looking at the totals, the results showed that visualised worked examples were preferred by at least 73% of the participants. And, a greater number of participants felt that visualised worked examples were more effective for learning, were more satisfied with them, and would select them for future worked example study.

4.2 Reasons for Selection

The most common theme that emerged among the reasons given by participants for their selections was related to understanding. Participants mentioned that the content was easy or easier to understand. Participants who selected both labelled and visualised designs mentioned this reason. Some did not provide further details as to what aspect was easier to understand. However, many of the participants elaborated further as discussed below.

Among those who selected visualised design, they reported that they were able to understand "what the purpose was for each section of the code" (S23). Some participants explicitly mentioned that the program was easier to understand because the visualised design related the subproblem in the analysis section to the program statements: "helped me understand the [program] lines which [relate] respectively to the analysis steps" (S21). This meant that the participant was able to understand which program statements were linked to a particular subproblem. This was made possible because of the highlighting feature, which resulted in both the subproblem and the program statements associated with it to be highlighted simultaneously. The purpose of the simultaneous highlighting was to help participants realise that those programs statements were addressing that particular subproblem. Since the subproblem specified what was required to be done, it served as an explanation of the purpose of highlighted program statements. By looking at the program statements linked to the subproblem, the participant may have been assisted in understanding what those program statements were achieving.

Table 1: Frequency distribution of students' selections for each question by groups and in total

			Worked Example Design			
			Labelled	Visualised	Labelled	Visualised
No.	Selection Criteria	Group	N (%)	N (%)	Total (%)	Total (%)
1	preferred overall	A	5 (35.7)	9 (64.3)	10 (26.3)	28 (73.7)
		B	5 (20.8)	19 (79.2)		
2	more effective for learning	A	2 (14.3)	12 (86.7)	5 (13.2)	33 (86.8)
		B	3 (12.5)	21 (87.5)		
3	more satisfied with	A	4 (28.6)	10 (71.4)	8 (21.1)	30 (78.9)
		B	4 (16.7)	20 (83.3)		
4	preferred for future worked example study	A	2 (14.3)	12 (85.7)	7 (18.4)	31 (81.6)
		B	5 (20.8)	19 (79.2)		

One participant elaborated that the program was easier to understand because the highlighting was "done part-by-part" (S30). The highlighting feature may have helped participants focus their attention on the relevant parts of the program, and to do so, a part at a time. This may have helped reduce the cognitive processing they needed because their concentration was directed to one small part at any one time. For participants who selected labelled design, the reason given for being able to understand the program was "it is easier to understand as the explanation is [written] above the code" (S26). So, for both designs, participants felt that they were supported in understanding the program because the different parts of the program were labelled (for labelled design) or highlighted (for visualised design) so they could understand each part. This resonates with findings from experimental studies of labelled worked example design (e.g., (Margulieux & Catrambone, 2016; Morrison, Margulieux, & Guzdial, 2015)). This is also consistent with the explanation of how signalling helps to draw attention to relevant information and to recognise the organisational structure of the information presented in learning materials (Lemarié et al., 2008; Lorch et al., 2011).

Whereas participants who selected labelled design restricted their comments about support for understanding for the program only, the participants who selected visualised worked examples went on further to point out others things that they found were easier to understand. One participant mentioned that it helped "understand what is the question asking" (S15). So, visualised worked example may have drawn the participants attention to look at the problem in addition to the solution program. This is an indication that visualised worked example may help participants carefully examine the problem that the program was solving. It is pointed out that none of the participants who selected labelled version mentioned about the problem. This implied that their focus mainly on the program.

A few participants who selected visualised design elaborated that it helped them understand the links between the problem, the analysis, and the solution program: "with the help of the analysis part, I can see clearly on the program which relates to the question's needs" (S17). Since the elements in the problem specification were highlighted simultaneously with the subproblem in the analysis section as well as the program statements, the links between these three different sections were made visible to the student. So, it may have helped participants to cognitively process these connections and relate all these elements together. It may have helped them understand the purpose of problem analysis and how problem analysis involved identifying the problem requirements. They may have been assisted in understanding how those requirements were fulfilled by the program statements. This finding is similar to studies on signalling where signals are used to show connections between related elements in learning content to help students to integrate the information (e.g., (Lemarié et al., 2008)). A few participants who selected visualised design reported that they had an understanding of the programming process (or coding) because of the analysis section: "Because it consists of the part of analysis which give[s] me a clear information [about] what should I do in coding" (S29). The visualised worked example design had a separate analysis section to emphasise problem analysis. These participants' comments about the analysis section showed that participants did take note of the analysis section, which may have made them aware of its purpose and importance. If students do not analyse a problem adequately, they may jump too quickly to design a solution. They may then realise that they lack sufficient information about the problem, as was observed in a study by Loksa and Ko (2016). The introduction of a problem analysis section in visualised worked example design may assist students to be more aware of problem analysis, and its role and importance, in the programming process.

A small number of participants reported that visualised worked example made it easier for them to find the relevant information: "easy to find the info from the problem […] easy to find the relevant code" (S4). Furthermore, a small number of participants reported that it helped them process the information in a shorter time. This meant that visualised design make it easier and faster to search for relevant information, which is stated as one of the benefits of signalling (Lemarié et al., 2008).

Some participants who selected visualised design mentioned that they found it was user friendly and it was suitable for beginners. These two aspects of the design were not mentioned as reasons for labelled design. On the other hand, three participants who selected labelled design mentioned that is was a presentation style that they were familiar with. Furthermore, one participant who selected labelled design mentioned that it was more appropriate to his learning method because he wanted "to read the question and solution and then analyse it myself" (S8). In other words, the participant felt that the highlighting was not necessary. Similarly, another participant felt that the highlighting feature was not necessary because he wanted to process the information for himself. These comments implied that the participants were probably knowledgeable and confident about their ability to process the worked example content, without the additional information provided by the visualised design. This suggested that students' self-efficacy or confidence in processing the learning content had an impact on the design they found were suitable for them. This is aligned to Moreno's (2006) suggestion that individual student differences, such as prior knowledge, may have an impact on the effect of example-based learning.

Another reason given by a participant who selected labelled design was that the visualised design required shifting attention to different sections to relate the information. Whereas, for the labelled design, since the comments were embedded in the program itself, it was easier to see the related information. Another two participants who selected labelled design did not like visualised design because it required moving the mouse to activate highlighting. As one of them mentioned, it was "troublesome to hover to the analysis [section] to show me the parts" (S37). The other mentioned that he wanted to take down notes, so having to move the mouse did not help. These reasons given by participants showed that the personal style of learning had an impact on the which design students preferred. This, again, suggested that consideration of the impact of student differences on worked example design is an important area of research (Moreno, 2006). The findings of the current study were are summarised in Table 2. The negative comments were not included.

Table 2: Summary of students' positive perceptions of both worked example designs

Labelled Worked Example Design	Visualised Worked Example Design
Understand the solution Analyse the information themselves Familiar presentation style	Understand the solution, the problem, the relation between problem and solution, and the programming process through analysis See or perceive the analysis, the relation between problem and solution, and the programming process Find needed information Suitable for beginners User friendly

Limitations of the current study are that it was conducted during a single class session and covered only one programming topic. Future research should consider other programming topics and the use of worked examples throughout the course. Another limitation is the small number of participants in the study. But, the preliminary findings of the current study could be used in the future as input for a more comprehensive questionnaire survey of students' perceptions of worked example design. A threat to the validity of the findings could be bias of the students toward visualised worked example design because of its novelty in terms of interactivity. But the negative comments of the participants about whether highlighting was necessary showed that participants were able to express their comments freely. Furthermore, since there were a number of participants who selected labelled worked example design as well showed that participants were at liberty to select either designs.

5. Conclusion

The current study compared labelled and visualised worked example designs, in the context of teaching and learning introductory programming, from the students' perspectives. The findings showed that more students favoured visualised worked example design. Although both designs led to cognitive engagement in terms of understanding the solution in the worked examples, the visualised design was also found to help in understanding the problem, the relationship between the problem and the solution, and the programming process. This may have been facilitated through the explicit listing of subproblems in the analysis section and the use of text highlighting in response to students' actions (i.e., interactive signalling). However, student differences, such as prior knowledge, may have an impact on which design students preferred, as shown by the negative comments about the highlighting in visualised design. Furthermore, emotional factors, such as familiarity with a certain design, and level of self-efficacy are factors to be considered in the design as well. The findings of this study provide preliminary results on students' perceptions, which may be used for future more comprehensive investigations into the different aspects of engagement with content, for better worked example design.

References

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. https://doi.org/10.1037/0022-0663.95.4.762

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General*, *127*(4), 355-376. http://doi.org/10.1037/0096-3445.127.4.355

Denny, P., Becker, B. A., Craig, M., Wilson, G., & Banaszkiewicz, P. (2019, July). Research this! Questions that computing educators most want computing education researchers to answer. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 259-267). https://doi.org/10.1145/3291279.3339402

Hanks, B., & Brandt, M. (2009). Successful and unsuccessful problem solving approaches of novice programmers. *ACM SIGCSE Bulletin*, *41*(1), 24-28. http://doi.org/10.1145/1508865.1508876

Johnson, B. and Christensen, L. (2014) *Educational Research: Quantitative, Qualitative, and Mixed Approaches*, 5th ed., SAGE Publications.

Lemarié, J., Lorch Jr, R. F., Eyrolle, H., & Virbel, J. (2008). SARA: A text-based and reader-based theory of signaling. *Educational Psychologist*, *43*(1), 27-48. https://doi.org/10.1080/00461520701756321

Liaw, S. S. (2008). Investigating students' perceived satisfaction, behavioral intention, and effectiveness of e-learning: A case study of the Blackboard system. *Computers & Education*, *51*(2), 864-873. https://doi.org/10.1016/j.compedu.2007.09.005

Loksa, D., & Ko, A. J. (2016, August). The role of self-regulation in programming problem solving process and success. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (pp. 83-91). ACM. https://doi.org/10.1145/2960310.2960334

Lorch, R., Lemarié, J., & Grant, R. (2011). Signaling hierarchical and sequential organization in expository text. *Scientific Studies of Reading*, *15*(3), 267–284. http://doi.org/10.1080/10888431003747535

Margulieux, L. E., & Catrambone, R. (2016). Improving problem solving with subgoal labels in expository text and worked examples. *Learning and Instruction*, *42*, 58–71. http://doi.org/10.1016/j.learninstruc.2015.12.002

Margulieux, L. E., Catrambone, R., & Guzdial, M. (2016). Employing subgoals in computer programming education. *Computer Science Education*, *26*(1), 44-67. https://doi.org/10.1080/08993408.2016.1144429

Mathieson, K. (2012). Exploring student perceptions of audiovisual feedback via screencasting in online courses. American Journal of Distance Education, 26(3), 143-156. https://doi.org/10.1080/08923647.2012.689166

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, *62*(2), 77-90. https://doi.org/10.1109/TE.2018.2864133

Moreno, R. (2006). When worked examples don't work: Is cognitive load theory at an Impasse? *Learning and Instruction*, *16*(2 SPEC. ISS.), 170–181. http://doi.org/10.1016/j.learninstruc.2006.02.006

Morrison, B. B., Margulieux, L. E., & Guzdial, M. (2015). Subgoals, context, and worked examples in learning computing problem solving. *International Computing Education Research Conference (ICER)*, 21–29. http://doi.org/10.1145/2787622.278773

Morrison, B. B., Margulieux, L. E., Ericson, B., & Guzdial, M. (2016, February). Subgoals help students solve Parsons problems. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 42-47). https://doi.org/10.1145/2839509.2844617

Nugroho, M. A., Setyorini, D., & Novitasari, B. T. (2019). The role of satisfaction on perceived value and e-learning usage continuity relationship. *Procedia Computer Science*, *161*, 82-89. https://doi.org/10.1016/j.procs.2019.11.102

Ozcelik, E., Arslan-Ari, I., & Cagiltay, K. (2010). Why does signaling enhance multimedia learning? Evidence from eye movements. *Computers in Human Behavior*, *26*(1), 110–117. http://doi.org/10.1016/j.chb.2009.09.001

Peart, D. J., Rumbold, P. L., Keane, K. M., & Allin, L. (2017). Student use and perception of technology enhanced learning in a mass lecture knowledge-rich domain first year undergraduate module. *International Journal of Educational Technology in Higher Education*, *14*(1), 40. https://doi.org/10.1186/s41239-017-0078-6

Prunuske, A. J., Henn, L., Brearley, A. M., & Prunuske, J. (2016). A randomized crossover design to assess learning impact and student preference for active and passive online learning modules. *Medical Science Educator*, *26*(1), 135-141. https://doi.org/10.1007/s40670-015-0224-5

Renkl, A. (2014). Toward an instructionally oriented theory of example-based learning. *Cognitive Science*, *38*(1), 1-37. https://doi.org/10.1111/cogs.12086

Schneider, S., Beege, M., Nebel, S., & Rey, G. D. (2018). A meta-analysis of how signaling affects learning with media. *Educational Research Review*, *23*, 1-24. https://doi.org/10.1016/j.edurev.2017.11.001

Schreier, M. (2012). *Qualitative Content Analysis in Practice*. SAGE Publications.

Selby, C. C. (2015, November). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the workshop in primary and secondary computing education* (pp. 80-87). https://doi.org/10.1145/2818314.2818315

Skudder, B., & Luxton-Reilly, A. (2014, January). Worked examples in computer science. In *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148* (pp. 59-64). Australian Computer Society, Inc..

Smith, A. R., Cavanaugh, C., & Moore, W. A. (2011). Instructional multimedia: An investigation of student and instructor attitudes and student study behavior. *BMC Medical Education*, *11*(1), 38. https://doi.org/10.1186/1472-6920-11-38

Yuriev, E., Naidu, S., Schembri, L. S., & Short, J. L. (2017). Scaffolding the development of problem-solving skills in chemistry: guiding novice students out of dead ends and false starts. *Chemistry Education Research and Practice*, *18*(3), 486-504. https://doi.org/10.1039/C7RP00009J