Evaluating the Efficacy of Peer-Created Worked-Example Videos in a Computer Systems Course*

Grace Kim, Dylan Green, and Suzanne J. Matthews
Department of Electrical Engineering & Computer Science
United States Military Academy
West Point, NY 10996
[grace.kim, dylan.green, suzanne.matthews]@westpoint.edu

Abstract

Worked examples are an educational tool widely used in introductory computer science classes, primarily for programming and code-tracing concepts. Prior research supports the use of worked examples as a scaffolding mechanism to help students build a solid foundation before tackling problems on their own. Whether breaking down the intricacies of code or explaining abstract theoretical concepts, worked examples offer a structured approach that nurtures a deeper understanding during self-study. This study explores how *peer-created* worked examples, shown through detailed step-by-step videos, aid student learning in an intermediate-level computer science course, namely computer systems.

Our results suggest that worked-example videos are a useful study aid for intermediate computer science courses, such as computer systems. Students who watched the worked-example videos found them to be very helpful, and ranked them as the top study aid for succeeding on quizzes. Additionally, students with access to worked-example videos performed moderately better on quizzes compared to students without worked-example videos. Our results and experiences also suggest that worked-example videos are beneficial to the students who created them as well as their peers who use them.

1 Introduction

Worked examples (also known as worked-out examples or worked solutions) are a pedagogical technique widely employed in teaching computer science. Formally, worked examples contain some formulation of a problem, some information on how to derive the solution, and the final answer [1]. Worked examples bridge the gap between the task and the answer, providing a solid and accurate foundation for students to later practice solving problems on their own [16, 10]. Worked examples have been widely cited as effective tools in the initial stages of learning procedural concepts in a wide variety of subjects[10], including algebra[14], chemistry[7], and english[5]. Students relied on these worked solutions to practice problems while reassuring themselves that they understood the necessary skills and underlying concepts [3].

Worked examples come in many forms, including text-based static examples, such as solution explanations in textbooks that are presented statically and all at once; or modeling examples, in which a teacher or peer generates a solution in real-time, allowing learners to see the solution built step by step; or dynamic examples, in which a custom tool, software, or animation presents a step-by-step solution of a problem of a code trace either using a custom tool or through an animation [9].

In the field of computer science, researchers have primarily delved into the impact of worked examples on introductory programming. As such, the efficacy of worked examples that illustrate code-tracing examples and program-building are widely studied [9, 12]. Skudder [12] describes worked examples in computer science as a “signature pedagogy"; however, most research on worked examples focus on instructor-created worked examples [8, 13, 6, 15, 4]. Prior work suggests that while students appreciate worked example videos containing code demonstrations, there was no statistically significant effect on student learning.

Research also suggests that this phenomenon is not restricted to worked examples that cover programming-only concepts. A recent study by Zavgorodniaia et. al [15] studied the effect of worked-example videos that diagrammatically explained Dijkstra's algorithm on a population of undergraduates who were primarily non-majors. The researchers found that access to the videos did not have a statistically significant effect on student learning, supporting an earlier result by Morrison [8].

Another recent study [4] performed a qualitative analysis on student perceptions of instructor-created video recordings of lectures, in which instructors presented static examples and live coding examples to students in an introductory computer systems course, and surveyed students on their thoughts on the two techniques. We note students had no control over the pacing of the videos, and could not pause them once started. The researchers reported that students found value in both modalities, liking the "at their own pace" studying and "focus on finished product" enabled by static worked examples, but the insights on instructor reasoning and "development process" of the live coding. In all the aforementioned cases, the videos were created by instructors, largely to control for high quality.

This paper looks at the impact of worked-example videos in a computer systems course at West Point, a four-year baccalaureate college. Our work is novel for several reasons. First, we evaluate *peer-created* worked example videos on student performance and perceptions, rather than instructor-created videos. Second, in addition to standard worked examples of program building and tracing, a non-trivial number of the produced worked-example videos involve non-programming content, such as reverse engineering (where learners observe how an assembly program translates to C code), cache address mapping, and visualizing process execution. Thus, our work adds to the body of knowledge on the effectiveness of worked example videos for non-programming content.

The rest of the paper is organized as follows. Section 2 provides an overview of our methodology, including details on the course implementation the worked-example video creation process, and experimental setup. Section 3 discusses the results of our quantitative and qualitative analyses. Lastly, we offer some reflections on lessons learned and major conclusions in Section 4.

2 Methodology

Data was collected over two fall offerings of CS380, a computer systems course taken at West Point, typically during junior year. The course generally represents students' first exposure to computer systems topics and is required for all computing majors at West Point. CS380 is split up into four units: C, Assembly, Memory & Code Optimization, and Parallel Computing. Concepts discussed in class are primarily evaluated through ten quizzes that are spread out over the semester, with one to three quizzes given every unit. The remainder of the course grade is determined by a series of multi-week projects, designed to build students' programming and assembly reading skills.

The investigators of this study include two undergraduate students and a faculty advisor. The undergraduate investigators took CS380 during Fall 2022 during their junior year, alongside their classmates. While enrolled in the course, they independently worked with the faculty advisor to create worked-example videos. The following semester, the students completed IRB training and worked with their advisor to design the study, which was reviewed and authorized by West Point's IRB process; the full study was executed in Fall 2023. The two students received independent study credit for the semesters they worked on the project.

Table 1 shows the spread of quizzes, in-class-exercises (ICEs) and corresponding worked-example videos produced throughout all of the units in CS380. Nearly half (42.5%) of the videos were created for the C unit, partially because this is the unit that students have traditionally had the greatest amount of difficulty in the course, and because it contains the largest number of corresponding in-class exercises.

Table 1: Number of Worked-example Videos in each Unit

Unit Number and Name	Num. Lessons	Num. Quizzes	Num. Videos	Num. ICEs
Unit 1: C	9	2	17	20
Unit 2: Assembly	10	3	9	14
Unit 3: Memory & Code Opt.	9	2.5	8	17
Unit 4: Concurrency	10	2.5	6	8
Total	38	10	40	59

2.1 Creating the Worked-Example Videos

For several years prior to (and including) Fall 2022, the CS380 course has incorporated worked examples, primarily in the form of modeled example solutions to in-class exercises (ICEs). Typically, after some amount of lecture, students are given time in class to complete a series of exercises related to the lesson's content, normally organized as a series of lesson worksheets. The instructor then works out the solution to one or more of the examples live in class. The following lesson, a static copy of the worked-out solutions of the majority of in-class exercises is distributed to students. Distributing solutions in this manner ensured students had access to solutions to in-class exercises that an instructor may not have had time to demonstrate in class. During a typical course execution, students are also told that the quizzes in the course borrow heavily from the concepts covered in the in-class exercises, and that they should primarily focus on the in-class exercise worksheets (and their corresponding provided solutions) as a study aid. The in-class exercises (and their solution files) do not change from year to year.

Armed with this information, the two undergraduate authors created forty worked-example videos of select in-class exercises. All the videos were less than ten minutes in length, with 65% shorter than five minutes. Each worked-example video had a static counterpart, namely the instructor-created worksheet solution PDFs provided by the instructor to all students in CS380. The process to create the worked-example videos was as follows: each student author selected twenty in-class exercises to create videos of, based on their own

Table 2: Population Statistics

Semester	Population Size (N)	“Weak"	“Average"	“Strong"
Control	44	7	21	13
Test	50	8	32	7

experience of what content was particularly difficult, and from their conversations with their classmates. After selecting the in-class exercises to port, each student author re-did the corresponding in-class exercises, and reviewed the associated static solutions, making sure to understand the problems fully, and consulted the faculty advisor if they had any questions.

Using an iPad to record their voice and their screens, each student slowly walked through each in-class exercise, recording the step-by-step solving process. Once recorded, the students used iMovie, a free editing software, to edit and polish each video and submit it for verification to the faculty advisor. The final stages of editing and publishing the videos involved an iterative verification and editing process, where the advisor gave the student investigators feedback, and the students edited their videos until they were deemed appropriately accurate and detailed. This process was done for all forty worked-example videos, with each video covering a different in-class exercise.

2.2 Data Collection

Data was collected over two Fall semester offerings of CS380. The Fall 2022 semester served as the “control" semester, where none of the students surveyed had access to worked example videos. The “test" population was the Fall 2023 semester, where all students had equal access to the worked example videos. In both semesters, students had access to the static worked-example solution PDFs that have always been provided in CS380. Additionally, students in the control and test semesters had access to a series of short instructor-created videos, which mainly contained a summary of the associated readings; however, a small quantity of the videos (especially from the assembly and concurrency unit) included some worked examples of content covered in the textbook.

Table 2 depicts some details about the populations under study; please note that the two student authors who were enrolled in Fall 2022 are excluded from the control population, as they interacted with the worked-example video content, while their classmates did not.

As part of the semester preparation for CS380, the instructor normally looks at the set of incoming students and “flags" certain students based on their performance in the three pre-requisite courses. If a student earned C grades or lower in all their pre-requisite courses, they are flagged as someone who may struggle in the course. Similarly, if a student has earned A-grades in all the pre-requisite courses, they are flagged as someone who will typically do well. Noticing that the test population had a larger number of struggling students and half the number of strong students as the control population, we partitioned the control and test populations into performance categories for part of our analysis: students in the “strong" partition earned an A- or higher in the pre-requisite courses for CS380; students who struggled and earned a C or lower in their courses were placed in the “weak" partition. Lastly, all other students were placed in the “average" partition. The sum of the three partitions does not add up to the population size; foreign exchange students and non-majors (who did not take the pre-requisites at West Point) were excluded from the partitions, along with one student who repeated the course.

Quantitative and qualitative data were collected for this study. The quantitative data used for this study were primarily quiz grades in the control and test populations. The faculty author has taught CS380 for a number of years and therefore was able to ensure that the difficulty and outcomes tested by each quiz remained consistent between the control and test semesters. Videos were posted on the West Point's Microsoft Stream service; as such, we were also able to use the video view counts tracked by Microsoft Stream to get a rough idea of how often videos were being watched.

Students in CS380 in the test semester were also asked to take an optional, anonymous survey about their experiences with the worked-example videos in CS380, administered at the course midpoint and again at the end. To incentivize responses while maintaining anonymity, CS380 offered two points of extra credit for each survey if at least ninety percent of the students in the course filled out the survey. All surveys were distributed electronically with a consent coversheet that informed students of the risks and benefits of completing the study and included verbiage about the extra-credit incentive. Students had to explicitly consent before being able to view the survey questions.

The survey consisted of seven questions. The first question asked students to rank according to a 5-point Likert scale how helpful they found particular class resources (including the worked-example videos), ranging from “very unhelpful" to “very helpful". The second question asked students to rank (from “most important" to “least important") the study aids they used to succeed on quizzes. The next two questions asked students to self-report the number of worked-example videos they watched and what other not-listed resources students used to study for the course quizzes. The last three questions were open-ended response questions, that asked students what they liked about the worked-example videos, what they felt could be improved about the videos, and how impactful they felt it was to have access to videos created by a peer, vs. an instructor in the course.

3 Results

Quiz grades were tabulated over all 10 quizzes in CS380. All available quiz data for the Fall 2022 and Fall 2023 populations were used in this study, with the population statistics summarized in Table 2. Of the 50 students in the test population, 33 students responded to the mid-point anonymous survey (66% response rate), and 39 responded to the survey (78% response rate). Since the surveys were anonymous, it was impossible to associate responses with students in any of the partitions.

3.1 Quiz Performance

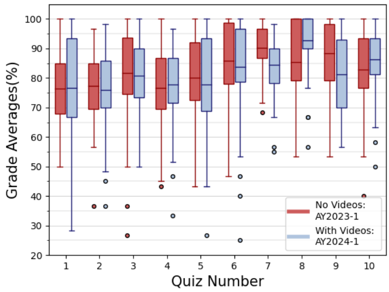
Figure 1 depicts box plots of the grades across all quizzes from the control and test semesters in CS380. Instead of the median, the average quiz grades are indicated by the middle line within each box plot. The box plots (along with the associated interquartile ranges and outliers) were generated using Python's Matplotlib `boxplot()` function.

Figure 1a shows the overall performance of the control and test semesters. As expected, the students in the control semester performed slightly better on the majority of the quizzes on average than those in the test population, owing to the larger number of characteristically weak students and the smaller number of strong students in CS380 in Fall of 2023. However, students in the test semester generally demonstrated slightly better performance on all the quizzes, with higher third quartile performance on three quizzes, higher first quartile performance on five quizzes, and either a higher "max" or "min" score on five quizzes.

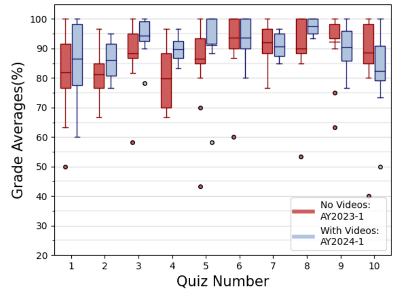
In the strong partition (Figure 1b), students with access to worked-example videos generally performed better than those who did not, having higher averages or third quartiles for six quizzes, and shorter interquartile ranges for five quizzes, suggesting consistently better performance. Interestingly, students in this group did worse than the control group on the last two quizzes, which covered concurrency topics.

In the average partition (Figure 1c), the students with the worked-example videos had first quartiles in six of the ten quizzes. In seven of the ten, the interquartile ranges are shorter in the test population, suggesting that students who had access to worked-example videos generally performed better on the quizzes. We note however that the means for both the control and experimental groups were very similar in the average partition, with differences of less than 4 percent.

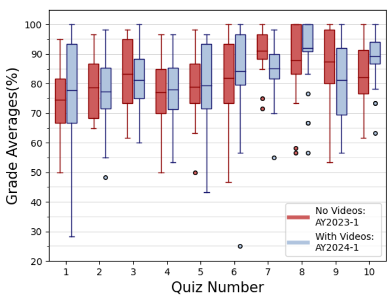
Lastly, in the weak partition (Figure 1d), the students in the test semester performed worse than the students in the control semester in seven of the ten quizzes, regardless of having access to worked solution videos. However, we do



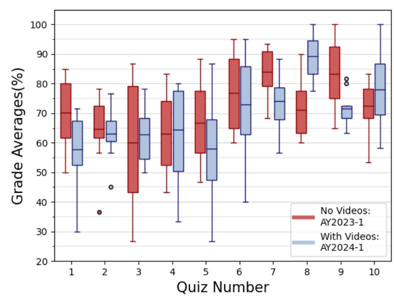
(a) Overall Average



(b) Strong Partition



(c) Average Partition



(d) Weak Partition

Figure 1: Box Plot Comparisons of Quiz Averages across Control and Test Semsters

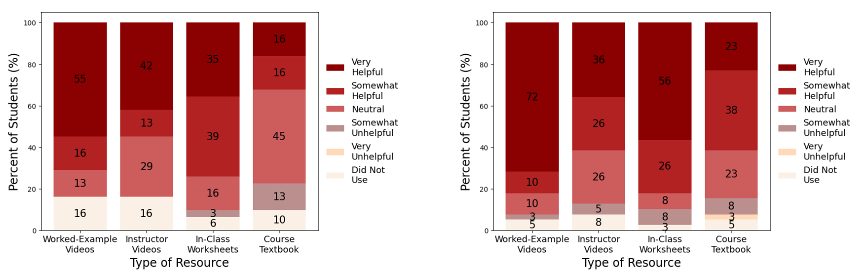
note that the interquartile ranges were generally smaller in the test semester than in the control semester, suggesting that students with access to worked-example videos in the weak partition had a lower variation in performance.

To better understand our results, we took a closer look at the video view counts and viewers associated with each video. All videos were watched by some fraction of the students; the high numbers of views suggested that students who watched the worked example videos watched them repeatedly, with viewership peaking immediately prior to a quiz. However, the data suggests that high view count of particular videos is primarily a reflection on student uncertainly of the material. Correlating view counts with quizzes suggest that the videos with the highest view counts were associated with quizzes that had some of the lower averages.

Quantitatively, it appears that the worked-example videos had the greatest positive impact on strong students and a modest positive impact on average and weak students. We note, however, that consistent with prior work, we did not see statistically significant differences in the means of the test and control populations.

3.2 Student Perspectives on Worked-Example Videos

To gain student perspectives on the helpfulness of worked-example videos, we asked students in the mid-point and course-end surveys to individually assess the helpfulness of the worked-example videos compared to other named class resources (the static in-class worksheet solutions, instructor-created videos, and the course textbook). We also asked students to rank the perceived usefulness of the aforementioned resources for studying for quizzes. Additionally, we asked them several open-ended questions to get a wide perspective of their answers.

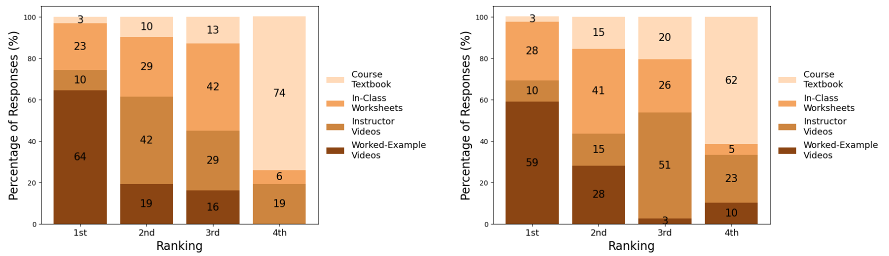


(a) Mid-Point Survey

(b) Course-End Survey

Figure 2: Student Perspective on Helpfulness of Worked-example Videos compared to other class resources

Figure 2 depicts how generally helpful students found worked-example videos and the other available classroom resources at the mid-point and the end of the course. In both the mid-point and course-end survey, the worked-example videos were consistently rated by a majority of students as “very helpful", with 55% at the mid-point and 72% at the course-end. We believe this is partially due to a lack of awareness of the worked-example videos on the part of some students; 16% of the students indicated that they did not use the worked-example videos in the mid-point survey, while only 5% indicated that they did not use the videos by the end of the semester. We note that the latter half of the course had fewer worked-example videos than the first half; correspondingly, a greater percentage of students reported the in-class worksheet solutions as “very helpful" in the course-end survey compared to the midpoint.



(a) Mid-Point Survey

(b) Course-End Survey

Figure 3: Student Ranking of Usefulness of Worked-example Videos compared to other class resources to succeed on quizzes

Figure 3 shows how students ranked the importance of each aforementioned classroom resource in helping them study and succeed on quizzes. In both surveys, the majority of students rated the worked-example videos as their top study resource, with a smaller fraction rating the static worksheet solutions as their top resource. Only 13% of students pointed to other resources as their top study aid. In short, our results demonstrate an enthusiasm for the video medium amongst the surveyed students.

3.3 Open Feedback

The final part of the survey given to the students of the course asked for open feedback on the worked-example videos. Students who reported watching the worked-example videos generally described them as being “thorough" and “well-explained", and that they appreciated the “step by step explanation" and “pace" of the explanations. *“I like how the videos explain thoroughly how to solve the problems and do not take any shortcuts"*, said one respondent. A few students also appreciated being able to pause and replay components of the worked-example videos: *“I can pause as much as I want and go at my own pace"* said one student.

Students did not offer much constructive feedback for improvement, except for requesting more worked-solutions videos. Some reported challenges accessing the videos due to unfamiliarity with the learning management system, which we plan to work on addressing in the future. A couple of students asked for explanations to be even “slower" and “more nuanced", highlighting the challenge of generating videos that appeal to all students: *“They are very good how they are"*, said one student, *“but sometimes they skip over a small* *section or step which may seem trivial, but when learning it is very useful to have each small step."*

Lastly, we asked students how impactful (if at all) having worked-example videos created by peers was rather than those created by instructors. The student respondents were fairly split; 38% felt that peer-created videos had a greater impact, and another 38% felt that it didn't matter. Students who felt that peer-created work-exampled videos were impactful alluded to how problems and concepts in the videos were explained in a way that were more intuitive to a student who is learning the material for the first time *"I believe the peers understand where other students minds are at and know we do not understand it as well as the instructors do,"* said one student. Students who preferred instructor-made worked-example videos alluded to the thoroughness and organizational qualities of the videos: *"instructor videos are much better planned, thought-out and organized,"* said a student. Another student succinctly stated the feelings of those that felt that the peer aspect did not matter: *"No impact. The fact that there was a video made all the difference."*

4 Lessons Learned & Conclusions

Our results suggest that worked-example videos are an effective study tool that moderately increased the average quiz score of students who used them as a study resource. Additionally, a majority of students rated worked-example videos as a helpful study resource, and also rated the videos as the most important study tool for quizzes overall. In addition to preferring the video modality, several students appreciated the peer-made nature of the videos, and asked for more to be produced. Prior work [11] suggest that the effects of peer-made solutions may also have farther-reaching effects than just performance in the immediate course; one study on the impacts of peer tutoring on tutor and tutee's performance found that the grade point average of tutees increased holistically rather than just in the course they received peer tutoring in [11]. We speculate that worked-example videos can function as a form of peer tutoring which would not only help students understand individual course subjects, but would help key students in on how the tutor thinks about course material. Individuals watching the videos can learn heuristics which translate to other computer science courses and thus increase their overall performance.

Our data also suggests that worked-example videos are an effective study tool for use in intermediate computer science courses such as computer systems courses. Having access to video worked-examples assisted student performance even in non-programming content in CS380, supporting the notion that worked-example videos are useful for a variety of non-programming topics in computing. In the semesters since they became available, the worked-example videos created by the student authors have become the most popular study resource in the course.

We conclude by offering some perspectives from the student and faculty authors on their experiences of the worked-example video creation process.

4.1 Student Creators: Experiences and Perspectives

While making the videos has continued to help the current student population, the act of *creating* the worked-example videos was incredibly formative for both student authors. We knew that these videos would be shared with peers in our department, and as such, we spent time to ensure that our thoughts were well laid out and concise. It was a time-intensive process, requiring over 80 hours of work to create, edit, and refine our videos.

Video deliverables require particular focus to create, and for the resource to be effective to others, it should be absent of erroneous content. Because this process was so lengthy and we had to understand the material at such a high level, the task of answering static questions on in-class quizzes, without the added pressure of narration and editing, was substantially easier. Quantitatively, we both ended the course with the highest letter grade, an A+, but more importantly, the process ensured we were incredibly confident in the material when we were quizzed.

Additionally, this study developed our understanding of basic video editing software, a skill that is translatable to other components of traditional college education such as group presentations and final projects. While not initially apparent, learning basic video editing skills has been helpful in a variety of different academic environments [2]. Furthermore, learning to teach and present material is a critical skill that not only improves an individual's understanding of the material but also develops critical presentation and interpersonal skills that are translatable across multiple disciplines.

4.2 Instructor Experience and Perspective

Based on prior work, the faculty author had two predictions: first, that having students create worked-example videos would be beneficial to their individual learning, and second, by offloading the work of video creation onto students, faculty time will be freed up to do other tasks.

In retrospect, having only two students (rather than the entire course) participate in the worked-example video creation process was important. As our surveys have shown, creating concise and well-explained worked-example videos is challenging, and requires some amount of effort. The faculty author spent quite a lot of time with the two students ensuring that video content was free of incorrect explanations and assumptions. That iterative process, while extremely valuable for the two student authors, was exceptionally time-consuming for the faculty author. In retrospect, it would have taken the faculty author less time to generate the worked-example videos on their own. However, the benefits to the students creating the videos (and to their peers who watched them) is compelling, and would have been undoubtedly lost.

There is an open question on the value of "crowd-sourcing" worked-example videos from the general student body during a particular course iteration. The perceived benefit would be that more students would have the opportunity to gain the insights offered through the video-creation process, like the students authors. On the other hand, creating good quality worked-example videos is time-consuming and difficult. The two student authors cared deeply about helping their peers and doing a good job; this is not always true of the average student. The more students involved in video production in a semester, the more faculty oversight that is required to make sure information is accurate. From the faculty author's perspective, the greatest challenge in producing high-quality peer-created videos is finding students who are willing and capable of producing high-quality study resources. West Point does not have undergraduate teaching assistants; the student authors however were popular tutors in the department.

Lastly, course designs involving well-established in-class problem sets like CS380 likely benefit the most from the worked-example video creation process, as it is a one-time operation whose products can be used over future semesters. For courses under active revision, creating high-quality worked-example videos that may ultimately be discarded might be judged as a poor use of time. However, our results show that the video modality for worked examples involving non-programming content is perceived as being very valuable to students, and that peer-created worked-example videos are rated as extremely valuable by both the peer creators and the student consumers alike.

Acknowledgement

This project is supported by the National Science Foundation (DUE - 2141814). The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense or the U.S. Government.

References

[1] Robert K Atkinson, Alexander Renkl, and Mary Margaret Merrill. "Transitioning from studying examples to solving problems: Effects of self-

explanation prompts and fading worked-out steps." In: *Journal of educational psychology* 95.4 (2003), p. 774.

[2] Christa Chewar and Suzanne J. Matthews. "Lights, camera, action! video deliverables for programming projects". In: *Journal of Computing Sciences in Colleges* 31.3 (2016), pp. 8–17. URL: https://dl.acm.org/doi/10.5555/2835377.2835380.

[3] Michelene T.H. Chi et al. "Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems". In: *Cognitive Science* 13.2 (1989), pp. 145–182. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1302_1.

[4] Derek Hwang et al. "A Qualitative Analysis of Lecture Videos and Student Feedback on Static Code Examples and Live Coding: A Case Study". In: *Proceedings of the 23rd Australasian Computing Education Conference*. ACE '21. Virtual, SA, Australia: Association for Computing Machinery, 2021, pp. 147–157. ISBN: 9781450389761. URL: https://doi.org/10.1145/3441636.3442317.

[5] Suna Kyun, Slava Kalyuga, and John Sweller. "The Effect of Worked Examples When Learning to Write Essays in English Literature". In: *The Journal of Experimental Education* 81.3 (2013), pp. 385–408. URL: https://doi.org/10.1080/00220973.2012.727884.

[6] Bridjet Lee and Kasia Muldner. "Instructional Video Design: Investigating the Impact of Monologue- and Dialogue-Style Presentations". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450367080. URL: https://doi.org/10.1145/3313831.3376845.

[7] Bruce McLaren and et al. "The Efficiency of Worked Examples Compared to Erroneous Examples, Tutored Problem Solving, and Problem Solving in Computer-Based Learning Environments". In: *Computers in Human Behavior* 55 (Feb. 2016), pp. 87–99. URL: https://www.sciencedirect.com/science/article/pii/S074756321530114X.

[8] Briana B. Morrison. "Dual Modality Code Explanations for Novices: Unexpected Results". In: *Proceedings of the 2017 ACM Conference on International Computing Education Research*. ICER '17. Tacoma, Washington, USA: Association for Computing Machinery, 2017, pp. 226–235. ISBN: 9781450349680. URL: https://doi.org/10.1145/3105726.3106191.

[9] Kasia Muldner, Jay Jennings, and Veronica Chiarelli. "A Review of Worked Examples in Programming Activities". In: *ACM Trans. Comput. Educ.* 23.1 (Dec. 2022). DOI: 10.1145/3560266. URL: https://doi.org/10.1145/3560266.

[10] Alexander Renkl and Robert K Atkinson. "Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective". In: *Cognitive Load Theory*. Routledge, 2016, pp. 15–22.

[11] Eun Hee Seo and Min Ji Kim. "The Effect of Peer Tutoring for College Students: Who Benefits More from Peer Tutoring, Tutors or Tutees?" In: *The New Educational Review* 58 (2019), pp. 97–106. DOI: https://doi.org/10.15804/tner.19.58.4.07.

[12] Ben Skudder and Andrew Luxton-Reilly. "Worked Examples in Computer Science". In: *Sixteenth Australasian Computing Education Conference (ACE 2014)*. Vol. 148. ACS. Auckland, New Zealand: Australian Computer Society (ACS), Jan. 2014, pp. 59–64. URL: https://dl.acm.org/doi/10.5555/2667490.2667497.

[13] Ben Stephenson. "Coding Demonstration Videos for CS1". In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, pp. 105–111. ISBN: 9781450358903. URL: https://doi.org/10.1145/3287324.3287445.

[14] John Sweller and Graham A. Cooper. "The use of worked examples as a substitute for problem solving in learning algebra". In: *Cognition and Instruction* 2.1 (1985), pp. 59–89. DOI: 10.1207/s1532690xci0201_3. URL: https://www.jstor.org/stable/3233555?seq=2.

[15] Albina Zavgorodniaia et al. "Algorithm Visualization and the Elusive Modality Effect". In: *Proceedings of the 17th ACM Conference on International Computing Education Research*. ICER 2021. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 368–378. ISBN: 9781450383264. URL: https://doi.org/10.1145/3446871.3469747.

[16] Rui Zhi et al. "Exploring the Impact of Worked Examples in a Novice Programming Environment". In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, pp. 98–104. ISBN: 9781450358903. URL: https://doi.org/10.1145/3287324.3287385.