Urhan, S., & Kocadere, S. A. (2024). The effect of video lecture types on the computational problem-solving performances of students. *Educational Technology & Society*, *27*(1), 117-133. https://doi.org/10.30191/ETS.202401_27(1).RP08

The effect of video lecture types on the computational problem-solving performances of students

Selin Urhan[*] and Selay Arkün Kocadere
Hacettepe University, Turkey // selin.urhan@hacettepe.edu.tr // selaya@hacettepe.edu.tr
[*]Corresponding author

(Submitted January 27, 2023; Revised June 7, 2023; Accepted June 16, 2023)

ABSTRACT: This study investigated the effect of video lecture types on the performance of students in computational problem-solving practices. A total of 19 university students participated in the computational problem-solving practices that mostly required declarative knowledge, and 22 university students participated in the computational problem-solving practices that mostly required procedural knowledge. The practices were implemented in the Algorithm and Programming course and the Computer Programming II course. Three video lecture types (instructor-whiteboard, instructor voice-handbook, instructor-slides) were used in both courses. The one-way repeated measures ANOVA test was employed to determine if there was a significant difference between the problem-solving performances of the students based on the video lecture type. In the Algorithm and Programming course that required mostly declarative knowledge, the problem-solving scores of the students were significantly higher after the instructor voice-handbook video practice than those after the instructor-whiteboard video practice. On the other hand, in the Computer Programming II course that required mostly procedural knowledge, the problem-solving scores of the students were significantly higher after the instructor-whiteboard video practice than those after the instructor voice-handbook video practice. The students showed higher performance in the video lecture types they preferred in both courses. The students listed the factors that affect their video preferences as (a) the effect of the presence of an instructor in the video lecture on their attention, (b) the efficiency of the video lecture in examining many and various examples in a limited time, (c) the opportunity provided by the video lecture to revise the content and procedure, and (d) the efficient presentation of the knowledge. It is recommended that an instructor should be present in the video that includes mostly procedural knowledge, while there is no need for an instructor in the video that includes mostly declarative knowledge regarding computational problem-solving activities.

Keywords: Computational thinking, Video lecture, 21st century skills, Online learning

1. Introduction

Computational thinking (CT) is an approach to problem-solving that requires the use of core computer science concepts and logic skills to transform a problem into a more easily understandable form and design a system that can be comprehended by others (Qualls & Sherrell, 2010). CT is a skill that focuses on analyzing a problem and making necessary abstractions to solve the problem (National Academies of Sciences, Engineering, and Medicine, 2010). In the CT process, humans structure the solution of a problem in a manner that can be efficiently performed by an information-processing agent (Wing, 2010). However, in this process, humans do not replicate the thinking mode of computers (Wing, 2006). They focus on how to solve problems using computers rather than working directly with computer hardware; this process does not need absolute use of a computer or machine (Wing, 2008).

It is stated that CT is a basic skill individuals should develop in the 21st century (e.g., Ma et al., 2021) as it is important for the development of other skills such as mathematical literacy and problem-solving (e.g., Cui & Ng, 2021; Korkmaz et al., 2017; Ng & Cui, 2021; Voogt et al., 2015). CT allows individuals to solve complex and challenging daily life problems by utilizing information and computing, enhancing their analytical thinking skills, and carrying out problem abstraction (Qiu, 2009). Students' high level CT skills allow them to be engaged in the learning process at high levels (Li et al., 2012). In addition, the computational principles and problem-driven approach also enhance students' interest in computing (Hambrusch et al., 2009).

CT can be taught within the context of various subjects, but it is preferred to be taught through programming (Lye & Koh, 2014) as students are directly exposed to CT when they engage in programming (Sabarinath & Quek, 2020). However, students often face challenges in reading and writing codes, tracing the codes in a systematic order, learning programming concepts and associating them, and writing programs (Xia, 2017). In order for students to learn programming (Sabarinath & Quek, 2020) and computing proficiently (Guzdial, 2008), teachers need to support their students with alternative teaching strategies in their studies. For this purpose, there have been studies carried about online teaching of CT (e.g., Hsu et al., 2018; Jocius et al., 2022; Liu et al., 2023; Monteiro et al., 2019; Zitouniatis et al., 2023). Online activities are indeed suitable to the nature of coding itself, the pace of individuals' acts during coding, and the progression speed (Mikkonen, 2019). In addition, online sessions can facilitate student learning by providing additional sample solutions, hints, and feedback (Milicic et al., 2020).

Educational videos facilitate online learning experiences as they offer the opportunity to present knowledge in both visual and audio formats, thereby enriching the learning experience (Chen & Wu, 2015), encouraging further involvement of learners in the learning process (Bruce & Chiu, 2015), and flexibility of scheduling and pace (Howard et al., 2018). The features of video lectures such as pause, rewind, and replay provide students with the opportunity to cover the content at their own pace and in their preferred time (Hong et al., 2018). Students can catch up on missed classes (Jung & Lee, 2015), study for exams (Bonafini et al., 2017; Traphagan et al., 2010), learn how to solve specific problems (Jung & Lee, 2015), and review challenging concepts (Bonafini et al., 2017) through video lectures. Some students prefer video lectures to traditional courses as they can choose the content, the learning environment, and the time (Hill & Nelson, 2011). It has been revealed that in recent years, students with different academic levels prefer online videos to improve their learning performance (Jung & Lee, 2015).

In the literature, some studies revealed that video lectures had a positive effect on the learning performance of students (e.g., Dalal, 2014). However, some studies reported that video lectures do not promote the learning process of the students (e.g., Pal & Patra, 2021), or the effect of video lectures on the learning performance is not significant or explicit (e.g., Kim & Chen, 2011). It is considered that one of the factors that leads to the ambiguous effect of video lectures on student performance and the learning process is the type of knowledge in the content. In this context, the literature suggests that the type of knowledge should be considered as a factor in the design of video lectures (Hong et al., 2018; Höffler & Leutner, 2007; Wang et al., 2020).

CT incudes both declarative knowledge, which involves learning and using ready-made commands in a programming language and procedural knowledge, which involves algorithm creation, loop structures, and program writing. Therefore, when designing video lectures during the CT instruction, it is important to consider the knowledge type. Some studies have been conducted on the opportunities provided by the video lectures in CT instruction, and it has been found that they offer opportunities for students to repeatedly watch the course content and make up for missed classes (Hsu et al., 2018). However, there haven't been any studies carried out yet about the use of different video lectures, considering the type of knowledge in the computational problem-solving practices, and comparing the outcomes in terms of online learning.

This study aimed to investigate the effect of different video lecture types on the computational problem-solving performances of students in online programming courses that include different type of knowledge. It is also aimed to reveal students' preferences for the use of different video lecture types in computational problem-solving practices. It is necessary to unveil the reasons why students favor or disfavor a specific video lecture type since instructors and information technology developers can understand how students perceive video lectures and can create and design desirable video contents that suit the specific needs of students (Pal & Patra, 2021; Shoufan, 2019).

2. Theoretical framework

2.1. Computational thinking

CT was initially introduced by Papert (1980) and popularized by Wing (2006). It is a problem-solving process that involves practices such as logical thinking, algorithmic thinking, abstraction, choosing the most appropriate strategy to solve a problem, and generalization (Computer Science Teachers Association & International Society for Technology in Education, 2011). Brennan and Resnick (2012) presented CT concepts that young children used during programming activities using a blockbased program as the following:

- Sequence: a series of individual instructions that can be executed by a computer to carry out a task,
- Loop: a mechanism that allows the same sequence to be repeated multiple times,
- Parallelism: the execution of different sequences of instructions simultaneously,
- Event: something that causes another thing to happen,
- Conditional: decision-making structures based on specific conditions,
- Operator: mathematical, logical, and string expressions that enable programmers to manipulate numbers and strings,

- Data: values stored, retrieved, and/or updated through variables or lists.

Brennan and Resnick (2012) defined the practices in which children were engaged during CT in the blockbased programming environment under four categories:
- Being incremental and iterative,
- Testing and debugging,
- Reusing and remixing,
- Abstracting and modularizing

In the context of being incremental and iterative, students structure the program in the series of small steps to accomplish a task. Testing and debugging requires to develop strategies to eliminate or diminish the problems that are detected in the program. This practice is related with reusing and remixing which requires to transfer something from the programs that were created by others or by getting support from someone who is experienced in programming. In the practice of abstracting and modularizing, programmers aim to build something extensive by combining small parts of the program together. This practice makes it easier for the programmer to think about different parts of the program, and easier for others to read and understand the program.

Weintrop et al. (2016) present a taxonomy that represents CT for mathematics and science and defined "computational problem-solving practices" as a category within this taxonomy. Computational problem-solving practices depend on the fact that enhancing students to understand scientific and mathematical events using programming can support them to improve their conceptual understanding about mathematical and scientific concepts.

Ng and Cui (2021) developed an analytical framework by combining the CT perspective of Brennan and Resnick (2012) which is related with child-friendly programming activities and the CT perspective of Weintrop et al. (2016) which is related with the intersection between CT and mathematical thinking practices. They combined the practices that are modeling, algorithmic thinking, debugging and troubleshooting in Weintrop's et al. (2016) taxonomy with the CT practices defined by Brennan and Resnick (2012) as abstracting and modularizing, reusing and remixing, and testing and debugging. Thus, Ng and Cui (2021) formed an analytical framework to analyze the CT processes of students in problem-solving practices. In this framework, the computational problem-solving practices are categorized as the following:
- Modeling: using the representations to construct original concepts,
- Abstracting and modularizing: elaborating the problem-solving process considering various details,
- Algorithmic thinking: solving a problem step by step,
- Reusing and remixing: constructing something with the help of other products or ideas,
- Testing and debugging: checking the procedure and finding out the problematic parts if there are.

In our study, computational problem-solving practices were used during each video lecture in the context of both programming courses. In each practice, the computational problems were used that require CT concepts (Brennan & Resnick, 2012) to be used and the CT practices (Ng & Cui, 2021) to be implemented to build the program that produces a solution for the given problem.

2.2. Video lecture design

According to Chen and Wu (2015), there are different online video lecture types such as voice-over presentation, lecture capture, picture-in-picture, and Khan-style video. The voice-over presentation video lecture includes the audio recording of a lecture and the slides that present the content. The lecture capture video lecture consists of the instructor, the whiteboard, and the presentation notes. The picture-in-picture type consists of the image and voice of the instructor and presentation slides. Khan-style videos include handwritten notes and the voice of the instructor.

Considering the Social Learning Theory, Wang et al. (2020) posit that learners might experience higher levels of satisfaction when they see the instructor in the video lecture and social cues may enhance the understanding of conceptual information. They state that the instructor in the video may provide social cues such as glance or facial expressions, which could lead to an interaction between the instructor and the learner and help the learner achieve deeper learning. The instructor in a video lecture attracts the attention of students and facilitate the teaching of both easy and challenging topics (Wang et al., 2020).

Guo et al. (2014) argue that the presence of the human face stimulates more intimate and personal emotions and prevents the monotonous aspect of presentation slides. It is reported that lecture capture and picture-in-picture video lecture types are more effective than the voice-over presentation as they can better increase the learning performance of students (Chen & Wu, 2015). Pi and Hong (2016) report that video podcasts including PowerPoint slides with instructors lead to enhanced learning. Kizilcec et al. (2014) state that most students have a better video lecture experience when the video includes the image of the instructor. These results indicate that the existence of an instructor in videos improves students' learning experiences.

According to the cognitive theory of multimedia learning, the voice-over video lectures might split learners' attention (Chen & Wu, 2015) and learners may have cognitive load due to processing the image of the instructor (Homer et al., 2008). Wilson et al. (2018) argue that when an instructor presents a video lecture, students' attention and understanding may be negatively influenced due to the visual features of the instructor.

Hong et al. (2018) analyzed the effect of the existence of an instructor in video lectures when the video lecture included declarative knowledge or procedural knowledge. Declarative knowledge refers to "know-what", while procedural knowledge refers to "know-how" (Schunk, 1996). If an individual acquires declarative knowledge, then the individual comprehends and remembers the knowledge. If the individual acquires procedural knowledge, then this person has the related declarative knowledge and is also conscious about how to use it in a task to reach the aim. Hong et al. (2018) found that the existence of an instructor in a video lecture supported the acquisition of declarative knowledge but increased the cognitive load of the students in the learning of procedural knowledge.

Some studies focusing on video lecture design in online education have included students' views on different video lectures. It was determined that students' actual learning performance and their views about the efficiency of these lectures may conflict (e.g., Wilson et al., 2018). It is crucial to evaluate video lectures from the student perspective (Shoufan, 2019) and to identify the conditions under which students are satisfied during different video lecture types (Nagy, 2018). In addition to experimental studies, students' views should be asked about the effectiveness of video lectures in the learning process to conduct a comprehensive analysis of the effectiveness of different video lecture types in the learning process.

3. Aim of the study

In this study, it is aimed to investigate the effect of different video lecture types on the acquisition of different type of knowledge in the context of computational problem-solving practices. Considering the discrepancy between students' learning performance and their views of the teaching process, it is significant and necessary to obtain students' views for the sake of evaluating the impact of video lectures thoroughly on computational problem-solving performance. Hence, it is also aimed to assess the effectiveness of different video lecture types from students' perspective. In line with these aims, the following research questions were addressed:

RQ1: Is there a difference in the problem-solving performance of the students in computational problem-solving practices based on the video lecture type used for presenting the educational content

- consisting mainly of declarative knowledge?
- consisting mainly of procedural knowledge?

RQ2: Is there a difference in the preferences of students in computational problem-solving practices for the video lecture type used for presenting the educational content

- consisting mainly of declarative knowledge?
- consisting mainly of procedural knowledge?

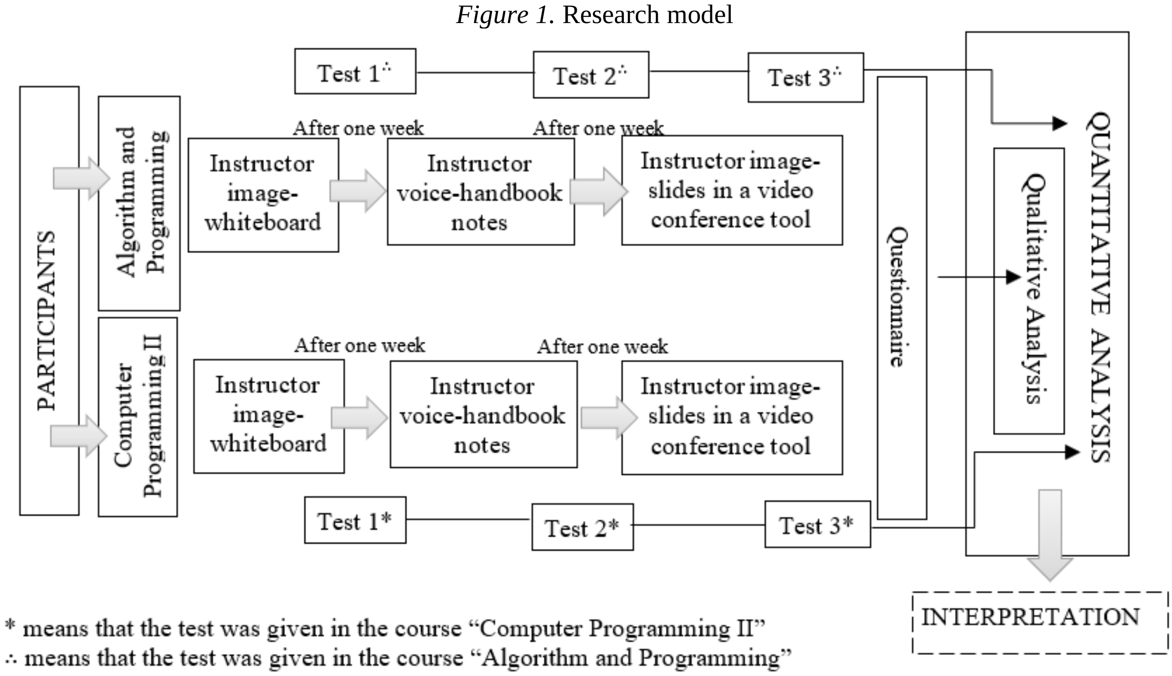
4. Methodology

4.1. Experimental design

The study utilized the embedded design, which is a mixed method design (Creswell, 2012). The quantitative part of the research was conducted using the repeated measures design to address RQ1. It was investigated if there was a significant difference between the problem-solving scores of students after each practice. The practices were implemented one week apart (see Figure 1). The achievement test was administered in the following week of each practice. The one-way repeated measures ANOVA test was used to determine the differences between the problem-solving scores of students. The qualitative part of the research was conducted to address RQ2. Content analysis was performed to reveal the preferences of the students regarding the video lecture type used for presenting the educational content. Finally, the results obtained from the quantitative and qualitative analyses were combined and interpreted (see Figure 1).

The participants were 41 university students enrolled in the Mathematics Education Program. Among the students, 19 were taking the Algorithm and Programming course, which mainly includes declarative knowledge about the principles of Maple commands applied in computational problem-solving practices. 22 students were taking the Computer Programming II course, which mainly includes procedural knowledge and covers the algorithm concept, flow diagram, and loop structures applied in computational problem-solving practices. Students in this study experienced learning through video lectures for the first time both throughout their college education in general and specifically in the context of computational problem-solving in Algorithm and Programming and Computer Programming II courses.

Figure 1. Research model



Three units with similar levels of difficulty were recorded for each course in three different video lecture types: (1) instructor image-whiteboard, (2) instructor voice-handbook notes, and (3) instructor image-slides in a video conference tool. The content covered in the video lectures is specifically related to the computational problem-solving, and it reflects the topics studied in the introduction of the Algorithm and Programming and Computer Programming II courses over three consecutive weeks. It was checked by the course instructor that the difficulty levels of the topics covered in the video lectures were parallel. In addition, the opinion of an instructor who is an expert in mathematics, mathematics education, and programming and who has taught computer programming courses for ten years was taken and it was assured that the subjects in the video lectures prepared within the context of each course were parallel and of the same difficulty level. Since Guo et al. (2014) indicated that shorter videos maintain students' attention and engagement, the video lectures were prepared as short videos of 15 minutes on average.

Algorithm and Programming and Computer Programming II courses were carried out through distance education, and all course materials, including lecture notes and presentations, were shared with students on the Moodle open-source learning platform. The video lectures were also made available to students on the Moodle open-source learning platform; the students were asked to watch each video within a week. At the end of each video lecture, the students were administered a test that assessed their problem-solving performance regarding the content of each video lecture. After all video-lecture presentations were completed, a questionnaire was administered to the students to allow them to evaluate the video lectures. In the questionnaire, the students were asked to make a preference order considering different types of video lectures and explain the reasons behind their preferences.

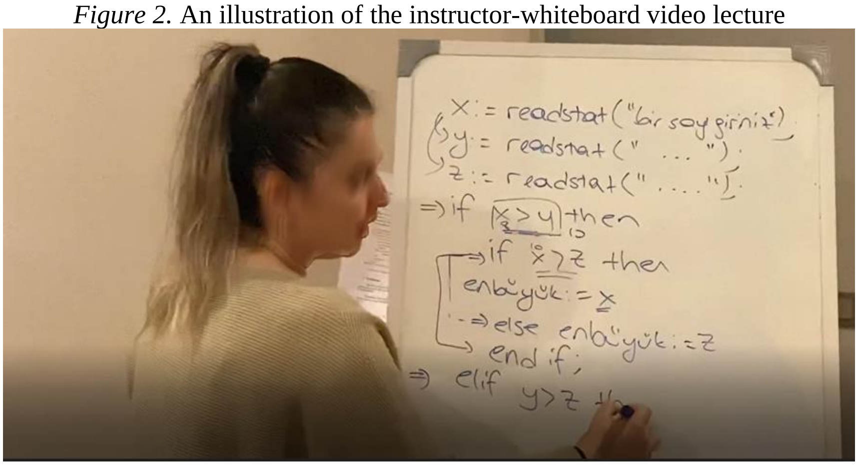
The researcher introducing the video lectures is the instructor for Algorithm and Programming and Computer Programming II courses. The role of the instructor in the presentation of these video lectures was to convey the content. The instructor also monitored whether students watched the video lectures within the given time interval. The instructor prepared exams that evaluated whether students have learned the content presented in each video lecture in the context of each course and the questionnaire consisted of questions that evaluated whether students have liked or disliked the video lectures. After each video lecture, the researcher as the instructor administered the exam and finally the survey to students online and then assessed students' answers and views.

4.2. Video lecture types

The instructor has structured each of the three video lecture types in a way that conveys the computational problem-solving process and the relevant content to the students. The three types of video lectures used in the study are as follows.

(1) The instructor-whiteboard video includes a traditional lecture. The content is presented by the instructor on the whiteboard. The instructor's voice and image, and the way of transferring instructions via writing the notes on the whiteboard are recorded simultaneously using a digital video camera for online viewing (see Figure 2).

Figure 2. An illustration of the instructor-whiteboard video lecture



Throughout this video lecture, the instructor stands next to the whiteboard, and she explains the content to the students by constructing the computational problem-solving process including the relevant content step by step and cumulatively. During this process, the instructor writes down each computational problem-solving step on the whiteboard, linking it to the previous step and, she provides verbal explanations to further elaborate and clarify the content, and thus making it more understandable for the students.

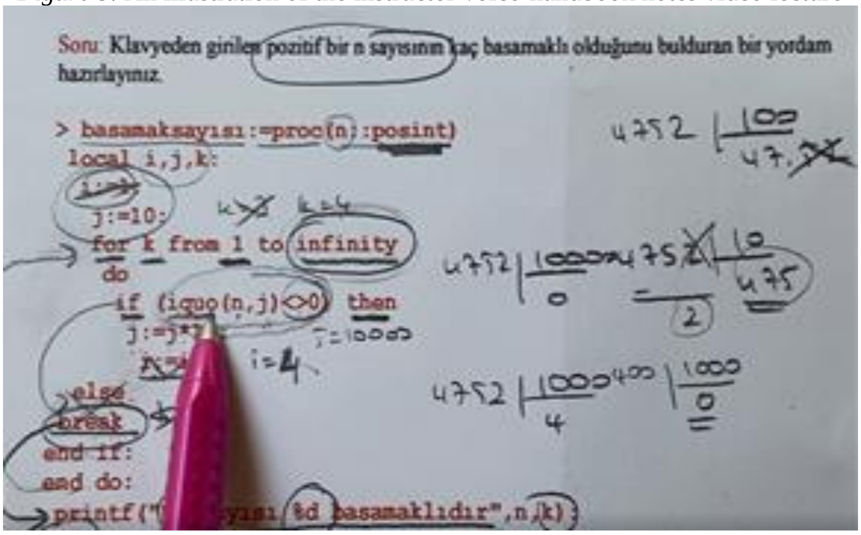
The instructor faces the whiteboard while writing down the content and simultaneously provides verbal explanations. The instructor sometimes turns towards the camera, looking at it while verbally explaining the content written on the whiteboard. Throughout the video, students can see the instructor's face, glances, and body movements, as well as hear her voice clearly. They can also clearly see the content written on the whiteboard. In this context, this video lecture type includes the instructor's written notes on the whiteboard as well as her voice and image.

(2) Instructor voice-handbook notes is a kind of voice-over video lecture and can be defined as a speech-based lecture. It includes book notes, a voice-over explaining the notes, and a pencil used by the instructor to point out the content presented (see Figure 3).

In this video lecture, the camera captures the computational problem-solving process that includes the relevant content from the book, displaying it step by step on the screen. This allows the problem-solving process to be built by the instructor in a cumulative manner, like the instructor-whiteboard video lecture. The instructor has conveyed each step by relating it to the previous one.

During this process, the pen the instructor uses to highlight and point out important parts of the content is seen on the screen and the instructor's voice is heard as well. In this video lecture, the instructor's image is not present. The instructor writes additional notes on the book, performs mathematical operations required in the problem-solving process, and underlines the parts she wants to emphasize. At the same time, the instructor clarifies the topic and enhances understanding by providing verbal explanations. In this context, the video lecture includes the instructor's book notes, which consists of written explanations made by the instructor on the book pages, verbal explanations, and the supplementation of the information in the book with voice-over explaining.

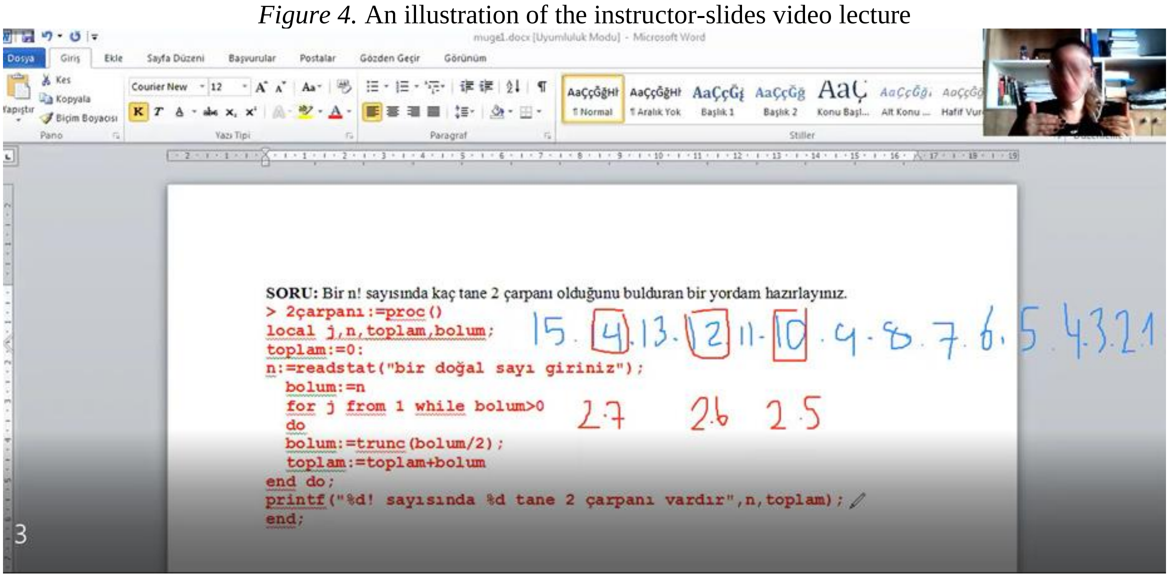
Figure 3. An illustration of the instructor voice-handbook notes video lecture



(3) The instructor-slides video lecture is recorded using a video conference tool and WebCam. It includes a combination of the image and voice of the instructor and the slides presenting the content to learners (see Figure 4).

The instructor uses the screen sharing feature of the video conference tool to display slides containing the content on the screen. Like the other two video lecture types, in this video lecture as well, the instructor builds the computational problem-solving process step by step and cumulatively, encompassing the relevant content. The image of the instructor is present in the upper right-hand corner of the slide, where the content is being presented. Students can see the instructor's face, glances, and hand movements, and hear her voice.

Figure 4. An illustration of the instructor-slides video lecture



The instructor can write additional notes on the slide that presents the content, make annotations, make marks, highlight the lines, and thus provide verbal explanations to the students in written form on the slide notes as well. In this context, this video lecture type includes a computer document containing the content, the written notes taken by the instructor, the instructor's image on a small screen and the instructor's voice.

4.3. Data collection tools

The instructor who prepared the video lectures developed achievement tests to evaluate whether the knowledge presented in the video lectures was acquired. The achievement tests included problems that required computational problem-solving practices, which are modeling, abstracting and modularizing, algorithmic thinking, reusing and remixing, and testing and debugging.

The achievement tests for the Algorithm and Programming course included problems that required declarative knowledge about the principles of Maple commands. The draft versions of the Test $1^{\therefore}$, $2^{\therefore}$, and $3^{\therefore}$ were revised by a mathematics educator who is an expert in mathematics education and programming other than the researchers considering the mathematical content and if they included parallel problems or not. The achievement tests for the Computer Programming II course included problems that required procedural knowledge. The students were expected to use for loop, if then comparison statement, while loop, and/or for-while loop to solve the problems. The draft versions of the tests were revised by the same expert considering both the mathematical content and if the problems included in Test 1*, 2*, and 3* were parallel or not.

After completing all video-lecture presentations, the students were asked to evaluate the video-lecture types. In the questionnaire, students were posed questions as “When you evaluate the video lecture types in terms of their effectiveness and efficiency in your learning process, what is your preference for the use of these techniques in your learning process? Mark the video lecture type you find most useful as 1, and mark the one you find least useful as 3”, and “Are there any conditions when your preference would change? Please explain.” Two experts were consulted regarding the suitability of the questions in the questionnaire to the purpose of the study.

4.4. Data analysis

The CT processes of the students in the achievement tests were analyzed based on the criteria prepared by Urhan (2022) based on CT framework of Ng and Cui (2021) in mathematics education. Hence, the computational problem-solving processes of the students were analyzed based on the requirements of rationality components (Boero, 2006; Morselli & Boero, 2009).

It is determined that all of the students watched the video lectures within the designated time, and hence CT processes of all the students are included in the analysis. The scores of the students were calculated out of 100. The one-way repeated measures ANOVA test was used to determine if there was a significant difference between the problem-solving scores of the students depending on the video-lecture type.

Content analysis was performed on the qualitative data collected through the questionnaire, and the reasons that determined the order of preference for video-lecture types were revealed. In order to clarify the parts that were not understood in the data obtained from the questionnaire, the relevant students were contacted and these parts were clarified with interviews. Hence, participant confirmation was obtained. In Section 5, in which quotations from the students are presented to the reader, the students that took the Algorithm and Programming course were coded as P1, P2, …, P19, while the students that took the Computer Programming II course were coded as P20, P21, …, P41.

5. Findings

This section presents the results regarding the performances of students within the programming courses with different contents, the preferences of students for video lecture types, and the reasons they provided for their preferences. First, the results regarding the video lecture types that presented the content of the Algorithm and Programming course are provided, followed by the results regarding the video lecture types that presented the content of the Computer Programming II course. Since the two courses are not equivalent in content, we did not compare the results for two courses.

5.1. The results regarding the algorithm and programming course

The one-way repeated measures ANOVA was administered to determine if there was a significant difference in the computational problem-solving performance of the students across the three different video lecture types in the Algorithm and Programming course. Normality of the data was assessed using the Shapiro-Wilk Test, which is more appropriate for small sample sizes (< 50 samples) (Field, 2009). Since the *p*-value was greater than .05, it was understood that the scores of students after each treatment were normally distributed (Field, 2009). The Mauchly's test was performed to assess sphericity. Since the *p*-value was .818 (> .05), we accepted the assumption that the variances of the differences between all possible pairs of within-subject conditions were equal. Thus, the assumption of sphericity has been met (Field, 2009). Table 1 presents the results of descriptive statistics for the independent variables.

Table 1. Descriptive statistics regarding the scores of students in the algorithm and programming course after each video lecture

	Mean	Std. deviation
Instructor voice-handbook	72.6316	18.88330
Instructor-slides	67.6316	18.13288
Instructor-whiteboard	61.0526	13.49680

The students obtained the lowest mean score in the instructor-whiteboard video lecture and the highest mean score in the instructor voice-handbook video lecture. The tests of within-subjects effects were performed to reveal whether there was an overall significant difference between the means for different video lectures. The results are presented in Table 2.

Table 2. Tests of within-subjects effects regarding the scores of students in the algorithm and programming course

Source		Type III Sum of Squares	*df*	Mean square	F	*Sig.*
Video type	Sphericity Assumed	1281.579	2	640.789	5.281	.010
	Greenhouse-Geisser	1281.579	1.954	655.763	5.281	.010
	Huynh-Feldt	1281.579	2.000	640.789	5.281	.010
	Lower-bound	1281.579	1.000	1281.579	5.281	.034
Error (video type)	Sphericity Assumed	4368.421	36	121.345		
	Greenhouse-Geisser	4368.421	35.178	124.180		
	Huynh-Feldt	4368.421	36.000	121.345		
	Lower-bound	4368.421	18.000	242.690		

Since the *p*-value was .010, it was deduced that the difference between the means was statistically significant [$F(2,36) = 5.281$]. Table 3 presents the results of the Bonferroni post hoc test, which enabled us to find the means that differed.

Table 3. The results of pairwise comparisons of the scores of students in the algorithm and programming course

Video type	(J) Video type	Mean difference (I-J)	Std. error	Sig.[b]
Instructor voice-handbook	Instructor-whiteboard	11.579*	3.312	.008
	Instructor-slides	5.000	3.785	.609
Instructor-whiteboard	Instructor voice-handbook	-11.579*	3.312	.008
	Instructor-slides	-6.579	3.608	.255
Instructor-slides	Instructor voice-handbook	-5.000	3.785	.609
	Instructor-whiteboard	6.579	3.608	.255

Note. Based on estimated marginal means. *The mean difference is significant at the .05 level. [b]Adjustment for multiple comparisons: Bonferroni.

It was found that the mean score of the students in the instructor voice-handbook video lecture was significantly higher than their mean score in the instructor-whiteboard video lecture ($p = .008$), but no difference was found between the mean score of students in the instructor voice-handbook video lecture and that of the instructor-slides video lecture ($p = .609 > .05$). In addition, the mean score of the students in the instructor-whiteboard video lecture was not significantly different compared to the mean score of the students in the instructor-slides video lecture ($p = .255 > .05$).

5.2. The results regarding the computer programming II course

The one-way repeated measures ANOVA was performed to determine if there was a significant difference in the computational problem-solving performances of the students when the content was presented with three different video lecture types within the Computer Programming II course. The Shapiro-Wilk Test showed that the *p*-value was greater than .05 and the scores of the students were normally distributed (Field, 2009). In order to assess sphericity, Mauchly's test was performed. Since the *p*-value was .742 (> .05), the assumption of sphericity has been met (Field, 2009). In Table 4, the results of descriptive statistics were presented for the independent variables.

Table 4. Descriptive statistics regarding the scores of students in the computer programming II course after each video lecture

	Mean	Std. deviation
Instructor-whiteboard	70.9091	14.11149
Instructor-slides	63.4091	25.51313
Instructor voice-handbook	55.4545	19.08060

It was seen that the students had the highest mean score in the instructor-whiteboard video lecture and the lowest mean score in the instructor voice-handbook video lecture. The results of the tests of within-subjects effects are presented in Table 5.

Table 5. Tests of within-subjects effects regarding the scores of students in the computer programming II course

Source		Type III Sum of Squares	*df*	Mean square	*F*	*Sig.*
Video type	Sphericity Assumed	2628.030	2	1314.015	6.083	.005
	Greenhouse-Geisser	2628.030	1.943	1352.709	6.083	.005
	Huynh-Feldt	2628.030	2.000	1314.015	6.083	.005
	Lower-bound	2628.030	1.000	2628.030	6.083	.022
Error (video type)	Sphericity Assumed	9071.970	42	215.999		
	Greenhouse-Geisser	9071.970	40.799	222.360		
	Huynh-Feldt	9071.970	42.000	215.999		
	Lower-bound	9071.970	21.000	431.999		

Since the *p*-value was .005, it was deduced that the difference between the means was statistically significant [$F(2,42) = 6.083$]. Table 6 presents the results of the Bonferroni post hoc test, which was performed to determine the means that differed significantly.

Table 6. The results of pairwise comparisons regarding the scores of students in the computer programming II course

Video type	(J) Video type	Mean difference (I-J)	Std. error	Sig.[b]
Instructor voice-handbook	Instructor-whiteboard	-15.455*	4.055	.003
	Instructor-slides	-7.955	4.507	.276
Instructor-whiteboard	Instructor voice-handbook	15.455*	4.055	.003
	Instructor-slides	7.500	4.707	.378
Instructor-slides	Instructor voice-handbook	7.955	4.507	.276
	Instructor-whiteboard	-7.500	4.707	.378

Note. Based on estimated marginal means. *The mean difference is significant at the .05 level. [b]Adjustment for multiple comparisons: Bonferroni.

The mean score of the students in the instructor voice-handbook video lecture was significantly lower than the mean score of the students in the instructor-whiteboard video lecture (p = .003), but no difference was found between the mean score of students in the instructor voice-handbook video lecture and that of the instructor-slides video lecture (p = .276 > .05). In addition, the mean score of the students in the instructor-whiteboard video lecture was not significantly different compared to their mean score in the instructor-slides video lecture (p = .378 > .05).

5.3. Differences in students' order of preference and the reasons for their preferences

5.3.1. Algorithm and programming course

In the Algorithm and Programming course, the students were asked to evaluate video lecture types in terms of their effectiveness in the learning process, and to order the lecture types based on their preference. The order of students' preference based on the effectiveness of video lecture types and the order of video lecture types in terms of students' problem-solving performance were consistent. The students preferred the instructor voice-handbook video lecture type the most, and the instructor-whiteboard video lecture type the least. The factors affecting their order of preference were grouped under the themes of attention, time, and type of knowledge. The themes are presented in Table 7.

Table 7. The factors affecting students' preferences of the video lecture types used in the algorithm and programming course

Themes	Quotations
Attention	"The writing of commands is very technical. Parentheses, name of commands, semicolon, parameters, etc. I could not catch these details on the whiteboard. But when I saw the written lecture notes, I did not get distracted. It was easy and quick for me to focus on the details." [P9]
	"I think the audio and lecture notes are sufficient. I listened directly to the instructor's voice and followed the lecture notes. I was distracted by the instructor's image near the notes in the video conference technique. I was absolutely lost in the whiteboard video lecture type. I could not catch the details. I had to study from the book after the lecture." [P3]
Time	"I think it is a waste of time when the instructor writes the commands on the whiteboard one by one. She just writes the commands. It is not something extra, nor a solution. It's like putting what is in the book on the whiteboard. Going through lecture notes is much faster." [P11]
	"After learning the usage of the commands, I want to see different examples of their usage. It is impossible for us to see so many examples on the whiteboard during one session of the lecture. Therefore, I did not find the whiteboard technique efficient." [P7]
	"It is much better to go through the lecture notes. In the book, we can both clearly see the spelling of the commands and quickly go over many examples. We do not have that chance on the whiteboard. Write-erase-write again; time is not enough." [P3]
Type of knowledge	"The content of this course is not like problem solving or proving, in which the result of one step must be compatible with another. In these kinds of courses, it would be better if the instructor structured the process on the whiteboard, but this lesson is not like that. Learning the rules to function the commands is enough to apply them." [P8]
	"The information in the book is to the point. It is enough to learn and remember the commands. There is no need for the instructor to write the commands and to make extra explanations on the whiteboard." [P14]
	"All content is about the application of some rules to function the commands. I do not need any extra explanation, note and/or drawing on the whiteboard to learn how to apply the rule. The instructor's voice over the lecture notes is sufficient." [P9]
	"After learning the usage of the commands, I want to see different examples of usage instead of more explanation about usage. Therefore, I did not find the whiteboard technique efficient." [P16]

As far as attention is concerned, the students stated that they could not catch the details and their attention was distracted after a while in the whiteboard technique. They found the audio narration over the lecture notes more useful as they could clearly see the technical writing of the commands. They stated that they could clearly see the details in the technical spelling of the commands in the lecture notes and they could focus on the details more easily and quickly. They emphasized that in the technique where the instructor explained the content on the whiteboard, they could not distinguish the details easily; therefore, they could not understand the subject completely.

As for time, the students stated that they wanted to examine many examples related to the use of the commands. They mentioned that the videos, in which the instructor gave a lecture through lecture notes, were more effective and efficient in terms of time management and progress in subjects. Hence, they preferred the videos, in which the instructor gave voice narration over lecture notes, both in order to see the spelling of the commands clearly and to examine many and various examples in a limited time.

Concerning type of knowledge, the students realized that declarative knowledge was predominant in the content of the course, and stated that the best learning tool for this information was the voice narration over the lecture notes. The students emphasized that the information required for command learning was presented in the exact flow in the book and it was unnecessary for the instructor to write this flow on the whiteboard. Therefore, they did not find the instructor-whiteboard technique efficient. They found the video lecture type, in which the flow was followed directly from the book and the declarative information was conveyed directly by the instructor's voice narration, more useful in this course.

5.3.2. Computer programming II course

In the Computer Programming II course, the students were expected to evaluate the video lecture types considering their effectiveness in the teaching process and make a preference order. The order of students' preference regarding the effectiveness of video lecture types and the order of video lecture types in terms of students' problem-solving performance were consistent. The students preferred the instructor-whiteboard video type the most, and the instructor voice-handbook video type the least. The factors affecting their order of preference were grouped under the themes of attention, revision opportunity, and type of knowledge. Table 8 presents the factors and some relevant quotations from student interviews.

Table 8. The factors affecting students' preferences for the video lecture types used in the computer programming II course

Themes	Quotations
Attention	"Seeing the instructor on the screen while listening to the topic makes me more alert. Just following the lecture notes make it difficult for me to focus. I get lost in the flow." [P22]
	"Seeing the instructor, her movements and gestures while she is explaining on the whiteboard creates a more dynamic environment. Listening to a recording as a voice note is very monotonous. I mean, it is boring. After a while, I get sleepy and stop following the lesson." [P30]
	"When I did not see the instructor in the audio recording on the lecture notes and video conference technique, I took a break very often. But I did not want to break the instructor-whiteboard video lecture. I wanted to watch until the end." [P38]
	"When I was going through the lecture notes of the book, I was not able to focus on the part where we created the main loop. Although the instructor tried to draw my attention to that part while explaining it through the lecture notes, the ready-made lines confused me." [P26]
Revision opportunity	"The instructor both gave an oral explanation and wrote on the whiteboard. So, it was like we went over it twice. This repetition made me understand the subject better." [P32]
	"I didn't take notes myself in the audio recording over the lecture notes and video conference technique, just because the book notes are in the book anyway. However, in the technique where the instructor narrates on the whiteboard, I recorded what she wrote on the whiteboard in my notebook and took additional notes for myself. I watched the video once again, checked my notes to see if there was anything missing. I can say that the whiteboard technique gave me a chance to revise." [P29]
Type of knowledge	"Creating a loop is not like applying a ready-made rule or using commands. It's like knowing the concepts and solving problems using them. I found it more useful when the instructor explained it on the whiteboard rather than seeing the loop as a pre-made rule in the book." [P33]
	"It's like problem solving or proving. I think it would be more understandable if someone structured the process and explained it." [P26]
	"The loop is something that is built step by step. It is necessary to decide what to do, what steps to take to reach the goal and to think step by step. It was very hard for me to think step by step in audio recording over the ready-made lecture notes." [P39]
	"Book notes prevented me from setting up the process in my mind. The video conference technique was not much different; the lecture notes were ready-made, but while the instructor was explaining the content on the whiteboard, I felt that we were building the process step by step." [P35]

Considering the theme of attention, the majority of the students stated that they could focus better and understand the subject more easily when they were listening to the content from the instructor and seeing the instructor simultaneously. The students emphasized that seeing the instructor allowed them to maintain their focus. They also mentioned that the instructor herself was a stimulant for them to focus on the flow, and they were lost in the video lecture conducted via audio recording over book notes as they could not see the instructor.

As for the theme of revision opportunity, in the video in which the instructor taught the subject on the board, the teacher's verbal and written explanation simultaneously enabled the students to better comprehend the subject and provided reinforcement. In the instructor voice-handbook and the instructor-slides in the video conference tool techniques, the students followed the notes while the instructor was speaking. However, they stated that it was not as effective and efficient for them in terms of understanding and repeating the content as following the notes that the instructor wrote on the whiteboard in her own handwriting.

As for type of knowledge, the students stated that the content of this course included procedural knowledge rather than declarative knowledge. Understanding procedural knowledge required detailed explanation and interpretation and creating loops. The students also emphasized that creating a loop or procedure also required creating small loops or comparison statements. They stated that they found the instructor-whiteboard technique more useful since they understood the task of creating piece-by-piece loops and then bringing them together on the whiteboard better. The explanations on the ready-made lecture notes made it difficult for them to decompose the loops.

6. Discussion and conclusion

Coding requires using both syntax rules and problem-solving strategies. Language syntax is the first thing to learn for programming and requires declarative knowledge. A deeper reflection on syntax facilitates the acquisition of procedural knowledge and the development of coding skills. In this study, the effect of video lectures on students' learning of declarative and procedural programming knowledge was investigated.

The study focused on two research questions: (1) Is there a difference in the problem-solving performance of the students in computational problem-solving practices according to the video lecture type used for presenting the educational content? and (2) Is there a difference in the preferences of students in computational problem-solving practices regarding the video lecture type used for presenting the educational content? The first research question aimed to determine whether students' problem-solving performances in computational problem-solving practices differed based on the video lecture. With the second research question, it was aimed to determine the effect of students' perceptions about the effectiveness of video lecture types on their computational problem-solving practices and to reveal the factors affecting their views.

Regarding the first research question, the students in the Computer Programming II course showed the highest performance in the video lecture where the instructor narrated on the whiteboard, followed by the instructor-slides in the video conference tool and the instructor voice-handbook notes technique. The students in the Algorithm and Programming course showed the highest performance in the instructor voice-handbook notes technique, followed by the instructor-slides in the video conference tool technique and the instructor-whiteboard video technique.

Regarding the second research question, when the effect of students' perceptions about the effectiveness of video lecture types on their computational problem-solving practices were examined, it was seen that they found the video lecture, through which they achieved highest performance, more beneficial, whereas they did not find the video lecture, with which they achieved lower performance, effective and efficient in terms of understanding the content of the course. Hence, in the Algorithm and Programming course, the students preferred the instructor voice-handbook video lecture type the most, and the instructor-whiteboard video lecture type the least, while in the Computer Programming II course the students preferred the instructor-whiteboard video type the most, and the instructor voice-handbook video lecture type the least.

Coding means the writing of computer programming code (Lye & Koh, 2014). According to Stephens (2018), programming can be defined as developing a logic-focused mindset by writing codes to record and execute algorithms in a formalized way. In this study, the Algorithm and Programming course, in which ready-made commands are taught, could be considered as coding, and the Computer Programming II, in which scripts are designed for a certain function, could be considered as programming. As stated by Mannila et al. (2014), we think that programming is an activity in which students perform more difficult tasks compared to coding.

Wang et al. (2020) reported that the presence of an instructor in a video lecture positively affects the learning of difficult topics. Ilioudi et al. (2013) also revealed that lecture capture was more effective compared to books for complex topics, and learning performance in lecture capture was higher than that in Khan-style video lecture. In our study, the students in the Computer Programming II course, which had a more difficult structure compared to coding in the Algorithm and Programming course, also became more successful after the instructor-whiteboard video lecture. Furthermore, the students who learned programming in the Computer Programming II course stated that the most suitable video technique for their learning process was the instructor-whiteboard video technique. The students, who learned to use ready-made commands in the Algorithm and Programming course, stressed that the content of the course had a mechanical structure that progressed in the form of input-output and required the use of declarative knowledge. They stated that the most suitable video lecture technique for this flow is the instructor voice-handbook notes technique.

It is reported in the literature that the presence of an instructor in a video lecture negatively affects students' attention (Wilson et al., 2018). In this study, the students who learned ready-made commands in the Algorithm and Programming course stated that they were distracted due to the visual presence of an instructor in the video lecture. Guo et al. (2014) stated that Khan-style videos were more engaging compared to PowerPoint slides. They reported that students' engagement improved when the video was shorter and the talking style was faster. In this study, the students in the Algorithm and Programming course found the instructor voice-handbook notes video technique more effective as it provided them with the opportunity to see more examples, ensured the transfer of knowledge with short, fast and clear explanations without wasting time on writing, presented them the technical usage and writing of commands in the clearest way, and did not include unnecessary verbal and written explanations.

As a result, the students found the technique, in which the instructor explained the content on the whiteboard, to be more useful in the lecture that included computational problem-solving practices based on procedural knowledge. However, the students found the video technique, in which the instructor was not present and in which they were exposed only to audio narration over the notes, to be more effective in the lecture which focused on the computational problem-solving practices containing mostly declarative knowledge. On the other hand, Hong et al. (2018) found that a video lecture that includes only the instructor facilitates the learning process of declarative knowledge, and cognitive load increases when students learn procedural knowledge. The study of Hong et al. (2018) was conducted in the context of an educational technology course, while this study focused on the computational problem-solving practices in mathematics. The difference between the results of the current study and the study of Hong et al. (2018) may depend on the topic taught.

It was seen that the effect of the presence of an instructor varied, depending on the type of knowledge taught in the instructional video focusing on the computational problem-solving practices. This result demonstrated that the selection of effective video types is influenced by the content. Similarly, Nagy (2018) argued that it is necessary to consider the content of the lecture to select the most effective video lecture type in the teaching process. In the context of computational problem-solving practices, it is recommended that the instructor should be present visually in the video lecture that includes mainly procedural knowledge. On the other hand, the audio-recording of the instructor over the lecture notes is more efficient and preferable in the video lecture that includes mainly declarative knowledge. Educators should consider these recommendations while designing online learning environments in the context of computational problem-solving since such a design would be consistent with the preferences of the students and also improve the problem-solving performance of students.

References

Boero, P. (2006). Habermas' theory of rationality as a comprehensive frame for conjecturing and proving in school. In J. Novotná, H. Moraová, M. Krátká & N. Stehlíková (Eds.), *Proceedings of the 30th Conference of the International Group for the Psychology of Mathematics Education* (pp. 185-192). PME. https://www.igpme.org/publications/current-proceedings/

Bonafini, F. C., Chae, C., Park, E., & Jablokow, K. W. (2017). How much does student engagement with videos and forums in a MOOC affect their achievement? *Online Learning, 21*(4), 223-240. https://doi.org/10.24059/olj.v21i4.1270

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (Vol. 1, pp. 1-25). https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Bruce, D. L., & Chiu, M. M. (2015). Composing with new technology: Teacher reflections on learning digital video. *Journal of Teacher Education, 66*(3), 272-287. https://doi.org/10.1177/0022487115574291

Chen, C. M., & Wu, C. H. (2015). Effects of different video lecture types on sustained attention, emotion, cognitive load, and learning performance. *Computers & Education, 80,* 108-121. https://doi.org/10.1016/j.compedu.2014.08.015

Computer Science Teachers Association & International Society for Technology in Education. (2011). *Operational definition of computational thinking for K-12 education.* https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Creswell, J. W. (2012). *Educational research planning, conducting and evaluating quantitative and qualitative research* (4th ed.). Pearson Education, Inc.

Cui, Z., & Ng, O. L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research, 59*(5), 988-1012. https://doi.org/10.1177/07356331209799

Dalal, M. (2014). Pilot impact of multi-media tutorials in a computer science laboratory course – An empirical study. *The Electronic Journal of e-Learning, 12*(4), 366-374. https://academic-publishing.org/index.php/ejel/article/view/1705/1668

Field, A. (2009). *Discovering statistics using SPSS* (3rd ed.). Sage Publications Ltd.

Guo, P. J., Kim, J., & Rubin, R. (2014). How video production affects student engagement: An empirical study of MOOC videos. *Proceedings of the First ACM Conference on Learning @ Scale Conference* (pp. 41-50). https://doi.org/10.1145/2556325.2566239

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25-27. https://doi.org/10.1145/1378704.1378713

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM Sigcse Bulletin*, *41*(1), 183-187. https://doi.org/10.1145/1539024.1508931

Hill, J. L., & Nelson, A. (2011). New technology, new pedagogy? Employing video podcasts in learning and teaching about exotic ecosystems. *Environmental Education Research, 17*(3), 393-408. https://doi.org/10.1080/13504622.2010.545873

Homer, B. D., Plass, J. L., & Blake, L. (2008). The effects of video on cognitive load and social presence in multimedia-learning. *Computers in Human Behavior, 24*(3), 786-797. https://doi.org/10.1016/j.chb.2007.02.009

Hong, J., Pi, Z., & Yang, J. (2018). Learning declarative and procedural knowledge via video lectures: cognitive load and learning effectiveness. *Innovations in Education and Teaching International, 55*(1), 74-81. https://doi.org/10.1080/14703297.2016.1237371

Howard, E., Meehan, M., & Parnell, A. (2018). Live lectures or online videos: Students' resource choices in a first-year university mathematics module. *International Journal of Mathematical Education in Science and Technology, 49*(4), 530-553. https://doi.org/10.1080/0020739X.2017.1387943

Höffler, T. N., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction, 17*(6), 722-738. https://doi.org/10.1016/j.learninstruc.2007.09.013

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126,* 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Ilioudi, C., Giannakos, M. N., & Chorianopoulos, K. (2013). Investigating differences among the commonly used video lecture styles. In *Proceedings of the Workshop on Analytics on Video-based Learning* (pp. 21-26). https://doi.org/10.13140/2.1.3524.9284

Jocius, R., O'Byrne, W. I., Albert, J., Joshi, D., Blanton, M., Robinson, R., Andrews, A., Barnes, T., & Catete, V. (2022). Building a virtual community of practice: Teacher learning for computational thinking infusion. *TechTrends, 66*(3), 547-559. https://link.springer.com/article/10.1007/s11528-022-00729-6

Jung, I., & Lee, Y. (2015). YouTube acceptance by university educators and students: A cross-cultural perspective. *Innovations in Education and Teaching International, 52*(3), 243-253. https://doi.org/10.1080/14703297.2013.805986

Kim, J., & Chen, C. Y. (2011). The influence of integrating pre-online lecture videos in classrooms: A case study. In C. J. Bonk & C. P. Ho (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 244-249). Association for the Advancement of Computing in Education. https://www.learntechlib.org/primary/p/38707/

Kizilcec, R. F., Papadopoulos, K., & Sritanyaratana, L. (2014). Showing face in video instruction: Effects on information retention, visual attention, and affect. *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 2095-2102). Association for Computing Machinery. https://doi.org/10.1145/2556288.2557207

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior, 72,* 558-569. https://doi.org/10.1016/j.chb.2017.01.005

Li, Z. Z., Cheng, Y. B., & Liu, C. C. (2012). A constructionism framework for designing game-like learning systems: Its effect on different learners. *British Journal of Educational Technology, 44*(2), 208-224. https://doi.org/10.1111/j.1467-8535.2012.01305.x

Liu, I. F., Hung, H. C., & Liang, C. T. (2023). A study of programming learning perceptions and effectiveness under a blended learning model with live streaming: Comparisons between full-time and working students. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2023.2198586

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41,* 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Ma, H., Zhao, M., Wang, H., Wan, X., Cavanaugh, W., T., & Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Educational Technology Research and Development, 69,* 1599-1616. https://doi.org/10.1007/s11423-021-10016-5

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In A. Clear & R. Lister (Eds.), *Proceedings of Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference* (pp. 1-29). ACM. https://doi.org/10.1145/2713609.2713610

Mikkonen, J. (2019). Bodygramming. Embodying the computational behaviour as a collective effort. *The Design Journal, 22*(1), 1423-1437. https://doi.org/10.1080/14606925.2019.1594967

Milicic, G., Wetzel, S., & Ludwig, M. (2020). Generic tasks for algorithms. *Future Internet, 12*(9), 1-16. https://doi.org/10.3390/fi12090152

Monteiro, A. F., Miranda-Pinto, M., Osório, A., & Araújo, C. L. (2019). Curricular integration of computational thinking, programming and robotics in basic education: A proposal for teacher training. In L. Gómez Chova, A. López Martínez & I. Candel Torres (Eds.), *Proceedings of the International Conference on Educational Research and Innovation* (pp. 742-749). https://doi.org/10.21125/iceri.2019.0232

Morselli, F., & Boero, P. (2009). Proving as a rational behaviour: Habermas' construct of rationality as a comprehensive frame for research on the teaching and learning of proof. In V. Durand-Guerrier, S. Soury-Lavergne & F. Arzarello (Eds.), *Proceedings of the Sixth Congress of the European Society for Research in Mathematics Education* (pp. 211–220). Institut National De Recherche Pédagogique.

Nagy, J. T. (2018). Evaluation of online video usage and learning satisfaction: An extension of the technology acceptance model. *International Review of Research in Open and Distributed Learning, 19*(1), 160-185. https://doi.org/10.19173/irrodl.v19i1.2886

National Academies of Sciences, Engineering, and Medicine. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press. https://doi.org/10.17226/12840

Ng, O. L., & Cui, Z. (2021). Examining primary students' mathematical problem-solving in a programming context: Towards computationally enhanced mathematics education. *ZDM - Mathematics Education, 53*(4), 847-860. https://doi.org/10.1007/s11858-020-01200-7

Pal, D., & Patra, S. (2021). University students' perception of video-based learning in times of COVID-19: A TAM/TTF perspective. *International Journal of Human-Computer Interaction, 37*(10), 903-921. https://doi.org/10.1080/10447318.2020.1848164

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books.

Pi, Z., & Hong, J. (2016). Learning process and learning outcomes of video podcasts including the instructor and ppt slides: A Chinese case. *Innovations in Education and Teaching International, 53*(2), 135-144. https://doi.org/10.1080/14703297.2015.1060133

Qiu, R. G. (2009). Computational thinking of service systems: Dynamics and adaptiveness modeling. *Service Science, 1*(1), 42-55. https://doi.org/10.1287/serv.1.1.42

Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges, 25*(5), 66-71. https://dl.acm.org/doi/10.5555/1747137.1747148

Sabarinath, R., & Quek, C. L. G. (2020). A case study investigating programming students' peer review of codes and their perceptions of the online learning environment. *Education and Information Technologies, 25*(5), 3553-3575. https://doi.org/10.1007/s10639-020-10111-9

Schunk, D. H. (1996). *Learning theories*: *An educational perspective* (2nd ed.). Prentice-Hall.

Shoufan, A. (2019). What motivates university students to like or dislike an educational online video? A sentimental framework. *Computers & Education, 134*(1), 132-144. https://doi.org/10.1016/j.compedu.2019.02.008

Stephens, M. (2018). Embedding algorithmic thinking more clearly in the mathematics curriculum. In Y. Shimizu & R. Withal (Eds.), *Proceedings of ICMI Study 24 School Mathematics Curriculum Reforms: Challenges, Changes and Opportunities* (pp. 483-490).

Traphagan, T., Kucsera, J. V., & Kishi, K. (2010). Impact of class lecture webcasting on attendance and learning. *Educational Technology Research and Development, 58*(1), 19-37. https://doi.org/10.1007/s11423-009-9128-7

Urhan, S. (2022). Using Habermas' construct of rationality to analyze students' computational thinking: The case of series and vector. *Education and Information Technologies, 27*(8), 10869-10948. https://doi.org/10.1007/s10639-022-11002-x

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728. https://doi.org/10.1007/s10639-015-9412-6

Wang, J., Antonenko, P., & Dawson, K. (2020). Does visual attention to the instructor in online video affect learning and learner perceptions? An eye-tracking analysis. *Computers & Education, 146*, 103779. https://doi.org/10.1016/j.compedu.2019.103779

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Wilson, K. E., Martinez, M., Mills, C., D'Mello, S., Smilek, D., & Risko, E. F. (2018). Instructor presence effect: Liking does not always lead to learning. *Computers & Education*, *122*, 205-220. https://doi.org/10.1016/j.compedu.2018.03.011

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2010). *Computational thinking: What and why?* [Unpublished manuscript]. Computer Science Department, Carnegie Mellon University. https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Xia, B. S. (2017). An in-depth analysis of learning goals in higher education: Evidence from the programming education. *Journal of Learning Design, 10*(2), 25-34. https://doi.org/10.5204/jld.v10i2.287

Zitouniatis, A., Lazarinis, F., & Kanellopoulos, D. (2023). Teaching computational thinking using scenario-based learning tools. *Education and Information Technologies, 28*(4), 4017-4040. https://doi.org/10.1007/s10639-022-11366-0