Comparison of the effects of two approaches to teaching conditionals with block-based programming in an online environment.

Hyunchang (Henry) Moon  
College of Education  
Texas Tech University  
hyunchang.moon@ttu.edu

Jongpil Cheon  
College of Education  
Texas Tech University  
jongpil.cheon@ttu.edu

Daniel Kelly  
College of Education  
Texas Tech University  
daniel.kelly@ttu.edu

Jaehoon (Jason) Lee  
College of Education  
Texas Tech University  
Jaehoon.lee@ttu.edu

Abstract

To date, while many educators and researchers have greatly argued for the importance of teaching computational thinking skills, there is a lack of research on an effective approach to CT education. This study aims to examine the effectiveness of two different approaches when teaching computational concepts with block-based programming language in an online environment. The study conducted a quantitative research using a pretest-posttest randomized-group design. Participants in the study were 101 undergraduate students enrolled in a computing and information technology online course. Two different teaching approaches (inductive vs. deductive) were delivered in an experimental web-based instruction during the spring semester of 2020. The knowledge test scores revealed that both groups were effective to increase the understanding of programming concepts. The inductive group outperformed the deductive group on the programming task of “repeat until”. Although the inductive group increased higher in the positive programming attitude than the deductive group, the change did not significantly differ between the groups. Lastly, the deductive instructional approach had a more positive attitude toward online instruction. Practical implication was further discussed.

Keywords: computational thinking, programming, online instruction, conditionals, block-based programming

Introduction

As programming requires cognitive and non-cognitive competencies, various pedagogical approaches should be considered. In a variety of subject areas, attempts to find effective pedagogical approaches have yielded significant advances in teaching and learning; however, block-based programming is not a common context in previous studies. Also, since block-based programming languages are taught to and played with individuals of all ages and at different levels of education, different approaches could be considered. For example, in order for helping novice learners increase the process of constructing key programming concepts, research is imperative to find out how to teach block-based programming effectively. In particular, conditional is one of the foundational computational concepts for controlling logical flow involved in programming and also necessary for solving
conditional problems. The conditionals usually involve the use of variables and operators to control the algorithmic flow, so it would make learners hard to acquire required knowledge and skills in conditionals (Duncan & Bell, 2015; Fronza et al, 2017; Grover et al., 2014). To facilitate learners to master the conditional statements of programming syntax, there is a need for studies to investigate effective approaches to teach the concepts in block-based programming environments.

Two distinct instructional approaches are divided into “inductive” and “deductive” (Felder & Silverman, 1988; Prince & Felder, 2006; Rieber & Parmley, 1995). Inductive and deductive teaching is an umbrella term that encompasses a range of instructional methods or strategies. The inductive approach proceeds from specific to general information. Conversely, the deductive approach goes from general and moves to specific. The two approaches have proven to be effective in other subject areas, such as language, mathematics, science, and chemistry. However, empirical evidence has not supported a consistent conclusion with regard to the effects of the inductive and deductive instructional approaches. Some studies reported inductive approach superior (De Jong, Acampo, & Verdonk, 1995; Kersh, 1958; Klauer, 1996; Schellhout et. al., 2006), while others reported findings in favor of deductive approach (Barrish, 1970; Erlam, 2003; Mohammaed et al., 2008; Owen, 2009). The other studies found the two approaches equally effective (Forgus & Schwartz, 1957; Shaffer, 1989). More importantly, no other studies exist examining the difference between inductive and deductive approach to teach block-based programming. Although deductive approach has been found typical to constructing new knowledge in other subjects (e.g., Atta, Ayaz, & Nawaz, 2015; Chiapetta et al, 1998; Prince & Felder, 2006; Murawska & Zollman, 2015; Shaffer, 1989; Singh & Yadav, 2017; Taha, 2014), inductive approach has not been applied to teach block-based programming. This study therefore focuses on how two different instructional approaches would help learners' cognitive and noncognitive aspects of programming in block-based environments.

Purpose of the Study

The main purpose of this study is to compare inductive teaching with deductive teaching approaches on students’ Computational Thinking (CT) knowledge achievement, programming skill, and attitudes towards CT and programming. Although the deductive teaching and learning has been found normal to constructing new knowledge in programming courses, the effect might depend on different context of learning and/or domain of learning; hence, inductive approach requires investigation. The inductive approach proceeds from specific to general information, whereas the deductive approach goes from general and moves to specific. Both approaches can offer certain advantages, but the biggest difference is the logic of reasoning. One way to advance this area of research is to compare the two different teaching approaches for learning a difficult computational concept, which can offer insight into instructional strategies. This study aims to examine the effectiveness of two different approaches when teaching undergraduate students about the construction of a computational concept. The conditionals (e.g., if/then, if/else, while conditional) are one of the difficult CT concepts. Also, these concepts are heavily related to other concepts (e.g., variable and operator) and also involved in constructing algorithms of programming. Therefore, meaningful understanding of these concepts by the students determined their CT competencies (i.e., knowledge and skill). In addition, this study explored the difference in online learners’ attitudinal perceptions towards the two instructional approaches.

This study investigated various learning outcomes of two teaching approaches (inductive vs. deductive) as follows: knowledge achievement, programming skill, and attitudinal perceptions. This study has the following research questions:

(a) RQ1: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ knowledge of conditionals?
(b) RQ2: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ programming skill of conditionals?
(c) RQ3: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ attitude toward block-based programming?
(d) RQ4: Does the student’s attitude toward instruction differ between two instructional approaches?

Theoretical Framework

Inductive and deductive teaching is an umbrella term that encompasses a range of instructional approaches. Inductive approach has the following characteristics: (a) Particular cases to general rules, (b) Concrete instance to
abstract rules, (c) Known to unknown, (d) Simple to complex; deductive approach, on the other hand, has the opposite (Singh & Yadav, 2017). Some studies reported the inductive approach superior (De Jong, Acampo, & Verdonk, 1995; Kersh, 1958, 1958; Klauer, 1996; Schelfhout et al., 2006); others reported findings in favor of the deductive approach (Barrish, 1970; Erlam, 2003; Mohammaed et al., 2008; Owen, 2009).

Method

The study conducted quantitative research approach, and pretest-posttest control-group randomized-group design.

Participants

Participants in the study was 101 undergraduate students enrolled in a computing and information technology online course at a large southwestern university during the spring semester of 2020. The participants enrolled from varied majors, were of various ages, and were both male and female.

Course Information

The course focused on a set of core competencies that shapes the background of computer science and also provided students with experience of learning Scratch block-based programming language. Out of 15 learning modules, nine modules were related to Scratch programming. In the learning module of conditionals, the two different teaching approaches (inductive vs. deductive) were delivered in an experimental web-based instruction.

Learning Content

The desired learning outcome was to create decision-making algorithms through the use of three conditionals (i.e., if/then, if/else, repeat until). Although the content of the topic studied in each group is the same, the sequence of the content was given in the reverse reasoning order to each group.

Instrumentation

Two groups’ CT competencies will be assessed from (a) Computational Thinking test (CTt; Roman-Gonzalez, 2015), (b) Dr. Scratch and Rubric programming artifact analysis (Moreno-León, Robles, & Román-González, 2015), and (c) Attitude survey (Authors).

Procedures

The pre- and post-test controlled group method is an effective design to minimize internal threats on experimental validity. The participants took a pre-test and pre-survey before taking the first module with Scratch. One group was taught on the topic of conditionals by the WBI with inductive teaching approach, whereas the other group was taught by the WBI with deductive teaching approach. The length of the WBI instruction took approximately three hours. The participants were randomly assigned to different conditions. They were asked to carefully read instruction on the computer screen and then followed the steps. After the online instruction, they took a post-test and post-survey. The Scratch artifact was collected in the assignment submission.

Analysis

The data was analyzed by T-Test or Analysis of Variance (ANOVA) to analyze the differences among group means in a sample. Also, multivariate analysis of variance (MANOVA) was used for comparing multivariate sample means. Moreover, repeated measures multivariate analysis of variance (RM MANOVA) was used to determine whether there were any differences in multiple dependent variables over times.

Findings

Researchers identified and coded significance statements pertaining to patterns and challenges in the coding journal responses. Codes were applied and revised as needed by combining and eliminating codes for parsimony. The final number of codes were counted for each category—being incremental and iterating, testing and debugging, reusing and remixing, abstracting and modularizing. According to Scratch coding journal responses, undergraduates’ reaction towards their programming experience considerably differed. The results are presented in the order of the research questions.

RQ1: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ knowledge of conditionals?
The result of multivariate test for time by instructional approach interaction was not significant (Wilks’ $\lambda = 0.93$, $F(3, 97) = 2.56$, $p = .06$, partial $\eta^2 = .07$). Instructional approach was significant for the score on the “Repeat until” conditional ($F(1, 99) = 6.43$, $p = .01$, partial $\eta^2 = .06$). Deductive teaching group made greater gains in the “Repeat until” score as compared to those in the inductive teaching group.

RQ2: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ programming skills of conditionals?

The total score was significantly lower for the deductive group ($M = 11.40$, $SD = 3.93$) than for the inductive group ($M = 13.10$, $SD = 2.77$) ($t(99) = -2.42$, $p = .01$, $d = 0.50$) and the difference was moderate. The score for the “If/then” conditional was significantly higher for the inductive group ($M = 4.53$, $SD = 1.14$) than for the deductive group ($M = 3.66$, $SD = 1.56$) ($t(99) = -3.21$, $p < .001$). The score for the “If/else” conditional was also significantly higher for the inductive group ($M = 4.57$, $SD = 1.06$) than for the deductive group ($M = 4.02$, $SD = 1.49$) ($t(99) = -2.13$, $p = .04$).

RQ3: Which approach, inductive or deductive, is more effective for enhancing undergraduate students’ attitude toward block-based programming?

Both groups made a positive change in the perceived programming attitude, which were statistically significant ($F(1, 100) = 6.94$, $p = .01$, partial $\eta^2 = .07$). The time effect was significant ($F(1, 100) = 6.43$, $p = .01$, partial $\eta^2 = .06$), but the interaction between time and instructional approach was not significant ($F(1, 100) = 2.93$, $p = .09$, partial $\eta^2 = .03$).

RQ4: Does the student’s attitude toward instruction differ between two instructional approaches?

The total score was significantly higher for the deductive group ($M = 3.69$, $SD = 0.84$) than for the inductive group ($M = 3.34$, $SD = 0.84$) ($t(99) = 2.09$, $p = .04$, $d = 0.04$). The score for Item 1 (“I like the online instruction”) was significantly higher for the deductive group ($M = 3.76$, $SD = 1.10$) than for the inductive group ($M = 3.27$, $SD = 1.15$) ($t(99) = 2.17$, $p = .03$). The score for Item 3 (“My interest in the subjects has been aroused.”) was also significantly higher for the deductive group ($M = 3.62$, $SD = 1.11$) than for the inductive group ($M = 3.06$, $SD = 1.24$) ($t(99) = 0.18$, $p = .02$).

Discussion and Implications

First, deductive approach produced a higher learning effect in terms of knowledge gain about conditionals, mainly when complex concepts or large numbers of facts are taught. Deductive approach is more effective for learning new conceptual information by making inference efficient to online college students. Second, inductive approach produced a higher learning effect in terms of programming skills of conditionals, particularly when guided practice is provided. Inductive approach is more effective for learning new programming skills by making generalization meaningful for online college students. Third, both the deductive and inductive approach produced a higher learning effect in terms of programming attitude. Inductive teaching made a greater positive change in programming attitude, although it is not statistically significant. Lastly, deductive approach made a more considerable positive effect in instruction attitude. Inductive approach may need more planning and resources to create clear guidelines and adequate scaffoldings.

The findings imply that both instructional approaches have proven to be useful when teaching programming competencies; however, both advantages and disadvantages exist between the two approaches. As a more diverse body of learners begins to learn block-based programming, educators need to consider the possible effects of different instructional approaches and incorporate them into their online programming instructions. For instance, appropriate Instructional approach should be adopted depending on learning objective(s), target learners, or level of complexity. Also, inductive approach needs to be more prepared to implement, and students’ understanding of learning topics should be assessed by a variety of measures. Moreover, tools and resources should be utilized for inductive approach to facilitate their learning process. Since both advantages and disadvantages exist between the two approaches, CS educators or instructional designers need to consider the possible effects of each instructional approach and incorporate them into their programming instructions. Although the study has some limitations, it
would offer insight into the direction of instructional methods or strategies for learners to have meaningful learning experiences in online programming courses.

Reference


