World Journal on Educational Technology: Current Issues

Volume 14, Issue6 , (2022) 1752-1764

www.wj-et.eu

Exploring college students' computational thinking in accounting spreadsheets design activities

Binti Muchsini *, Sebelas Maret University, Faculty of Teacher Training and Education, Education Science Department, Surakarta, Indonesia, https://orcid.org/0000-00002-6402-0583

Siswandari Siswandari, Sebelas Maret University, Faculty of Teacher Training and Education, Education Science Department, Surakarta, Indonesiahttps://orcid.org/0000-00003-4537-7939

Gunarhadi Gunarhadi, Sebelas Maret University, Faculty of Teacher Training and Education, Education Science Department, Surakarta, Indonesiahttps://orcid.org/0000-00002-1329-6280

Wiranto Wiranto, Sebelas Maret University, Faculty of Teacher Training and Education, Education Science Department, Surakarta, Indonesiahttps://orcid.org/0000-0001-5881-139Xc

Suggested Citation:

Muchsini, B., Siswandari, Gunarhadi & Wiranto (2022). Exploring college students' computational thinking in accounting spreadsheets design activities. *World Journal on Educational Technology: Current Issues*. *14*(6), 1752-1764. https://doi.org/10.18844/wjet.v14i6.7715

Received from July 20, 2022; revised from September 09, 2022; accepted from November 10, 2022.
Selection and peer review under responsibility of Prof. Dr. Servet Bayram, Medipol University, Turkey

Abstract

This study aims to investigate the extent to which computational thinking can be developed through constructionism-based accounting spreadsheets activities. This study design used a mixed-method approach, namely a participatory qualitative approach and a quantitative descriptive approach. Data were collected through documentation (college students' artefacts) and classroom observations. The results showed that constructionism-based accounting spreadsheets design can build and facilitate computational thinking development. The college students' emotional and social engagement when executing a design plan can foster curiosity and high enthusiasm to complete the design together. This engagement can reduce the cognitive load that students feel in understanding programming languages when utilising visual basic for application excel. This study contributes and suggests to learning practitioners to improve the students' quality so that they can compete in this digital era. This research can be used as a basis for conducting further research where researchers empirically investigate the impact of computational thinking development.

Keywords: Computational thinking, cognitive load, emotional engagement, accounting education;

* ADDRESS FOR CORRESPONDENCE: Binti Muchsini, Sebelas Maret University, Education Science Department, Surakarta, Indonesia,
E-mail address: binti_muchsini79@student.uns.ac.id

1. Introduction

Currently, the use of technology is a big challenge for the world of education (Niyazova et al., 2022). Spreadsheets are a technology that is widely used in the business world. Good spreadsheet skills are becoming a demand in the accounting profession (Frownfelter-Lohrke, 2017). Spreadsheets have been widely used and developed (Bock et al., 2020), but there are still many mistakes in designing spreadsheets, especially made by users who have minimal programming skills (Frownfelter-Lohrke, 2017). In business, this will have an impact on increasing the company's burden due to losses arising from user errors in the spreadsheets design.

Frequent errors in designing spreadsheets include domain errors, formula errors, spreadsheets design errors and typos (Schneider et al., 2017). These errors are also found among students majoring in non-computer science. Students majoring in non-computer science have a relatively higher potential for errors in designing spreadsheets than students majoring in computer science (Lawson et al., 2009). However, these errors can be minimised by developing a computational mindset for students majoring in non-computer science. Yeh, Xie, and Ke (2011) assert that computational thinking patterns, such as data representation, data processes, abstractions and procedural thinking, are needed in the utilisation of excel functions to produce correct and effective spreadsheet designs. Computational thinking is more than just how to use a computer and write code (Acevedo-Borrega et al. 2022). Currently, computational thinking is one of the new ways of thinking in helping to overcome the challenges of technological development (Israel-Fishelson & Hershkovitz, 2022). Therefore, it is very important for the education world to teach good spreadsheets design and skills by building a learning environment that facilitates students' computational thinking development.

Pedagogical strategies for teaching good spreadsheets design have been widely discussed and researched (Frownfelter-Lohrke, 2017; Schneider et al., 2017), but are not yet oriented towards computational thinking development. Theoretically and empirically, computational thinking can be developed through a constructionist learning environment (Gero & Levin, 2018). Therefore, this study offers a new way to teach good spreadsheets design by developing computational thinking through a constructionist learning environment. Papert's (1980) constructionism theory emphasises the learners' activeness in constructing artefacts rather than just getting information. This constructionism theory was further developed by Resnick (1996) which is known as distributed constructionism. The distributed constructionism theory examines three activities, namely discussion, sharing and collaborative construction (Resnick, 1996). This was supported by the argument of Hay and Barab (2001), which affirms that the learners' activeness in the constructionism learning environment can be seen when sharing and reflecting on artefacts collaboratively.

Gero and Levin (2018) have proven that computational thinking can be developed through the utilisation of spreadsheets in a constructionism learning environment. However, the study only developed one component of computational thinking, namely algorithmic thinking. This encourages researchers to implement constructionism-based accounting spreadsheets designing activities in order to develop computational thinking. This classroom-based research specifically aims to investigate the extent to which computational thinking can be developed through accounting spreadsheets design activities in a constructionist learning environment.

2. Literature review

2.1. Theoritical framework: constructionism as potential strategy to develop computational thinking

The theoretical framework of this study was constructionism. Constructionism was first introduced by Papert (1980) as a theory that encourages learners to acquire knowledge through actively constructing artefacts. The main aspect of constructionism was the learner has an 'object-to-think-with', for example, a 'turtle' is a digital animal in logo programming that acts as a means of thinking because it can be controlled and moved according to orders (Papert, 1980). The focus of constructionism lies in the active construction of knowledge by building artefacts (Girvan & Savage, 2019). Learners were active builders of their knowledge through the engagement of building artefacts (Butler & Lehay, 2021). The creation of artefacts or products allows the learner to gain an understanding of the concept studied in depth (Ostashewski et al., 2011).

The implementation of constructionism theory in learning can be carried out through the act of building artefacts in the framework of a process of representation of knowledge, such as software manipulation, or concrete actions, such as physical manipulations necessary to create artefacts (Ostashewski et al., 2011). Artefacts are objects for thinking together (Papert, 1980). Papert's (1980) constructionism theory was developed by Resnick (1996) which is known as distributed constructionism. Distributed constructionism is an extension of the constructionism theory that focuses on the engagement of more than one person in design and construction activities, namely discussing, sharing and collaborating in constructing artefacts (Resnick, 1996). This was supported by the argument of Butler and Lehay (2021:4), which emphasised that constructionism provides learning when others see and give criticism to the resulting artefacts. Thus, Resnick's (1996) constructionism criteria include the process of building artefacts such as discussing how to construct artefacts collaboratively and sharing artefacts with other groups.

2.2. Design principles of constructionism in learning spreadsheets

Spreadsheets are a medium for developing computational thinking, but spreadsheets cannot develop computational thinking by itself. Pedagogical activities play an important role in this case, namely the spreadsheets design activities to support students to develop a computational mindset. The designing spreadsheets activities include three activities: planning spreadsheets design, implementing the spreadsheets design plan and testing and documenting spreadsheets design (Schneider et al., 2017).

Based on Resnick's (1996) constructionism theory, spreadsheets design activities in this study can be described as follows:

a. Discussing artefacts or spreadsheet design, including the existing spreadsheet designs that will be targeted in preparing financial statements. The activities discuss spreadsheets design that already exists today, such as spreadsheets design observation activities obtained from schools or from social media, and identifying errors or deficiencies in spreadsheets design obtained from schools or from social media. Discussing spreadsheets design activities that will be targeted in compiling financial reports means planning spreadsheets design activities to correct errors or deficiencies of spreadsheets design obtained from schools or social media.

b. Collaboration in constructing includes executing spreadsheets design plans and testing the result of spreadsheets design. The executing spreadsheets design activities were planned in Microsoft Excel to solve financial reporting cases collaboratively. The testing spreadsheets design activities were carried out after spreadsheets design was completed to ensure whether the design created was accurate and automated.

c. Sharing spreadsheets design, including documenting the spreadsheets design process and presenting spreadsheet design to get criticism and suggestions from other groups.

The use of spreadsheets along with constructionism-based activities allows students to develop their computational thinking. This study will make an empirical contribution to the learning process of accounting spreadsheets. Table 1 shows the design principles of constructionism in the learning of accounting spreadsheets to develop the computational thinking of college students.

Table 1. Design principles of constructionism in learning spreadsheets

Constructionism criteria	Spreadsheets design activities	Design principles
1. Discussing artefacts	Planning spreadsheets design	Identify spreadsheets errors and then discuss spreadsheets design planning in groups.
2. Collaboratively constructing meaningful artefacts	Executing a spreadsheets design plan	Collaboratively executing the results of spreadsheets design planning that has been constructed using Microsoft Excel to solve financial reporting cases.
	Evaluating spreadsheets design	Testing and re-refining the spreadsheets design with teamwork so that it becomes accurate designs
3. Sharing artefact	Presenting spreadsheets design	Sharing the spreadsheets' design that has been documented in small groups in order to get criticism and suggestions from other groups so that interactions with other groups occur.

The constructionism theory states that the meaning construction occurs very well when the learner is involved in building and sharing artefacts; one of the learning concepts relevant to this theory is project-based learning because it involves the creation and presentation of products (Willis & Tucker, 2001). Project-based learning gives students the opportunity to work relatively independently over a long period of time, resulting in a realistic product or presentation (Jones et al., 1997). Project-based learning is learning that is able to answer the challenges of the 21st century. This is in line with the opinion of Basilotta Gómez-Pablos et al. (2017), who stated that project-based learning is an alternative strategy that effectively develops fundamental, critical and research skills. Projects were planned and managed involving complex and scheduled tasks (Sudjimat et al., 2021).

Project-based learning is a learning concept that organises learning around a project. A project is a complex task based on the challenging questions or problems that involves students in design, problem-solving, decision-making or investigative activities (Hsu et al., 2018). Thus, project-based learning strategies allow being able to improve students' skills in solving complex problems, including student computational thinking. This is in line with the opinion of Bower et al. (2017) that the improvement of computational thinking can be carried out by involving students in designing, problem-solving, decision-making or investigative activities, namely by involving students in project-based learning.

The constructionism theory can be used as a reference to improve the project-based learning concept that is currently underway in order to develop students' computational thinking. With reference to the constructionism theory, project-based learning in this study used the method of constructing spreadsheet designs together. The method of constructing the spreadsheets design together was implemented using the technique of small group discussions. Discussions in this small group were expected to facilitate students' computational thinking.

In the procedure for implementing small group discussions, students form small groups with four or five students. Each member brings the results of the analysis found during the spreadsheets design observation assigned by the lecturer (comparing the designs given by the lecturer or taken from social media with the spreadsheets design principles, followed by identifying the spreadsheets design errors based on the comparison results) individually and then discuss with the team.

The discussion results were used as the basis for making design planning that includes domain planning and spreadsheets planning. Domain planning is planning that is not related to spreadsheets, but planning related to the financial reporting process, namely policies and what reports are needed to compile financial statements. After creating a domain plan, planning spreadsheets should be created collaboratively including formula planning and by planning the inputs design, calculations and outputs. The next step was for students to collaboratively execute the plan made in the excel programme; then, test and refine the design to produce an accurate design; and the last step was to share the spreadsheets design with other groups for advice and input. Each spreadsheets design activity in constructionism learning is expected to develop students' computational thinking.

3. Method

3.1. Research design

This research used a mixed-method approach. Qualitatively, this study used a participatory qualitative approach to empirically investigate computational thinking that develops the class of spreadsheets naturally. The case study design was used to investigate the reality of spreadsheets design activities in a constructionist learning environment in order to develop computational thinking. The study used a participatory approach to increase students' engagement in the classroom through discussion, sharing and collaboration to produce an accurate spreadsheets design. Students were involved in making pedagogical intervention decisions in designing spreadsheets. Quantitatively, this study used a quantitative descriptive approach.

3.2. Participants

Thirty-eight college students in a university located in the eastern part of Java, Indonesia, participated in this research. 16 students came from vocational schools majoring in accounting and 22 students came from non-vocational schools. Students who came from vocational schools majoring in accounting had more experience in learning accounting compared to students who came from non-vocational schools. Therefore, the formation of a discussion group was carried out by dividing 16 students from vocational schools into 8 small groups first. Furthermore, 22 students from non-vocational schools were divided into 8 pre-formed groups.

3.3. Instructional procedures

The topic of accounting spreadsheets design was part of the spreadsheets course. This study was carried out over eight meetings. The time duration for one meeting was 100 minutes. College students' activities and computational thinking that will be developed in a constructionist learning environment can be seen in Table 2.

Table 2. Constructionism-based spreadsheets design activities

Stage	College students' activities	Assessment basis	Evolving computational thinking
Planning spreadsheets design (4 weeks)	Identifying spreadsheets design errors or flaws shared by lecturers	Notes on errors or flaws that were successfully identified	Abstraction
	Deciphering errors or flaws that have been identified in small sections	Notes on identified descriptions of errors or flaws	Decomposition

	Navigating and discussing various alternative solutions obtained from social media. Next, determining the relevant solutions to correct the error or deficiency.	Notes on alternative solutions from each group member	Generalisation
		Notes on the best solutions taken	Evaluation
	Drawing up measures based on solutions taken in stages and in detail.	Notes on the steps over which the solution is taken	Algorithmic
Executing spreadsheets design plan using excel (2 weeks)	Including a template of reports that were needed in order to compile financial statements on worksheets.	Design a template that supports the preparation of financial statements on worksheets.	Algorithmic
	Inserting formulas or codings on report templates in worksheets.	Formula design and coding on pre-designed templates in worksheets.	Algorithmic
Evaluating spreadsheets design (1 week)	Testing the design that has been created and re-refining the design if the design is not already working properly.	Spreadsheets design after repairs	Algorithmic
Sharing spreadsheets design (1 week)	Presenting design results and giving criticism and suggestions to other groups' designs	College students' criticism of the design spreadsheets of other groups	Abstraction
		Suggestions on spreadsheets design of other groups	Generalisation

Table 2 shows that the development of college students' computational thinking was carried out through four activities, namely planning, executing plans, evaluating and sharing collaboratively. The first stage (Planning spreadsheets design, 4 weeks) targeted the development of computational thinking on five components (abstraction, decomposition, generalisation, evaluation and algorithmic). In the second stage (Executing a spreadsheets design plan, 2 weeks), the development of computational thinking targeted through this activity was an algorithmic component. In stage three (Evaluating of spreadsheets design, 1 week), the targeted algorithmic components were developed through this activity. In stage four (Sharing spreadsheets design, 1 week), two components (abstraction and generalisation) were targets of computational thinking, with students providing criticism and suggestions to the design of spreadsheets of other groups.

3.4. Data collection and analysis

Data were collected through documentation and classroom observation. Documentation includes college students' artefacts (accounting spreadsheets design) and notes on the spreadsheets design planning process. College student-generated accounting spreadsheets design was used to analyse the extent of students' computational thinking skills developed through the stages of executing design plans and evaluating spreadsheets design. The spreadsheets design planning process report was used to analyse the extent to which students' computational thinking skills were developed through the stage of planning spreadsheets design. Class observation aims to analyse the extent to which students' computational thinking skills were developed through the stage of sharing spreadsheets design. Data was collected and described to obtain an idea of the extent to which the computational thought processes can be developed through the activity of designing accounting spreadsheets based on constructionism. In this study, quantitative analysis became the basis for describing the extent to which computational thinking was part of the process of designing accounting spreadsheets based on data collected through documentation. Meanwhile, qualitative analysis was based on data collected through class observation.

4. Finding and discussion

4.1. Planning spreadsheets design

The stage of planning the design of spreadsheets includes six activities that allow the development of computational thinking. The computational thinking developed through this stage can be seen in Table 3.

Table 3. Evaluation of computational thinking in planning spreadsheets design stage

Meeting	College Students' activities	Computational thinking component	Score (%) *n* = 8 groups			
			4	3	2	1
Meeting #1	Identifying spreadsheets design errors or flaws shared by lecturers	Abstraction	62.5%	25%	12.5%	-
Meeting #2	Deciphering errors or flaws into small parts	Decomposition	62.5%	25%	12.5%	-
Meeting #3	Exploring various alternative solutions from social media.	Generalisation	62.5%	25%	12.5%	-
	Discussing to determine the best solution from various alternative solutions	Evaluation	50%	12.5%	37.5%	-
Meeting #4	Making a domain plan by drawing up the steps over the solution taken in stages and in detail.	Algorithmic	62.5%	37.5%	-	-
	Making spreadsheets plan by drawing up steps on the solution taken in stages and in detail.	Algorithmic	50%	25%	25%	-

Table 3 shows that in the first activity 62.5% of the college student groups identified errors or deficiencies using the parameters of good spreadsheets design principles and company accounting cycle; 25% of the college student groups identified errors or deficiencies based solely on the parameters of the design principle alone or company accounting cycle alone; and 12.5% of the college student groups did not base identifying design errors or deficiencies on good spreadsheets design principles and company accounting cycles. These results show that most of the college student groups have been able to identify errors or deficiencies in the spreadsheets design shared by lecturers. In this first activity, students are guided to focus on spreadsheets errors, namely domain planning errors and spreadsheets planning errors. In addition, college students are also guided to focus on design deficiencies in terms of the process of preparing company financial reports. This activity supports the development of student computational thinking, especially abstraction. Tsai et al. (2020) affirm that abstraction is a method of solving problems by means of focusing on important information or ignoring unimportant complexities to facilitate problem-solving.

In the second activity, 62.5% of the student groups outlined errors or shortcomings into small parts using the parameters of good spreadsheets design principles and company accounting cycle; 25% of the student groups elaborated errors or shortcomings into small parts only based on the parameters of design principles alone or company accounting cycle alone; and 12.5% of the student groups did not base deciphering errors or shortcomings into small parts on good spreadsheets design principles and accounting cycles of the enterprise. These results also show that most of the student groups are already able to decipher the errors or deficiencies found in the first activity in detail and logically. This activity familiarises students with using a computational mindset, especially in the decomposition component. Bocconi et al. (2016) assert that decomposition is related to the ability to decompose complex problems into simpler problems or into smaller parts.

In the third activity, 62.5% of the college student groups made many alternative excel functions or coding to correct formula or coding errors based on spreadsheets design principles; 25% of the college student groups made a few alternative excel functions or coding to correct formula errors or coding based on spreadsheets design principles; and 12.5% of the college student groups made a little alternative excel or coding functions to correct formula or coding errors, but not based on spreadsheets design principles. These results show that most of the students are trying to provide many alternatives to correct errors or deficiencies in the previous spreadsheets design. In this activity, college students make alternative formulas or coding by navigating how to correct formula errors or similar identifiers through social media. This activity supports the development of computational thinking on the generalisation aspect. Tsai et al. (2020) affirm that generalisation is the ability to solve problems by means of using a specific pattern of problem solutions on similar problems.

In the fourth activity, 50% of the college student groups chose the relevant formula or coding from various alternatives offered to correct formula errors or coding based on spreadsheets design principles; 12.5% of the college student groups chose formulas or relevant coding from various alternatives offered to correct formula or coding errors, but not based on spreadsheets design principles; and 37.5% of the college student groups chose formulas or coding but not relevant to correcting formula or coding errors. These results show that half of the overall college student groups are able to make decisions on solutions that will be used to correct mistakes or shortcomings. This activity supports the development of computational thinking on the evaluation component. Tsai et al. (2020) affirm that evaluation is a skill in solving problems by finding the best solution by comparing available alternative solutions.

The fifth and sixth activities explore the solutions taken in detailed and gradual steps. In the fifth activity, 62.5% of the college student groups compiled financial reporting measures in stages, and in detail and 37.5% of the college student groups compiled financial reporting measures in stages, but not in detail. While in the sixth activity, 50% of the student group made formulas or coding gradually and in detail; 25% of the college student groups made formulas or coding gradually but not in detail; and 25% of the college student groups made formulas or coding but did not arrange it in stages or in detailed. The results of the analysis in the fifth and sixth activities show that most college students have been able to explore step by step the improvement plan for errors in stages and in detail. Computational thinking that can be developed through this stage is college students' engagement in algorithmic thinking. Selby and Woollard (2013) emphasised that algorithmic thinking is the ability to solve problems systematically and in detail.

4.2. Executing the spreadsheets design plan

The stage of executing the spreadsheets design plan includes two activities that allow the development of computational thinking. Computational thinking that can be developed through this stage is an algorithmic component (Table 4).

Table 4. Evaluation of computational thinking in executing the design plan stage

Meeting	College students' activities	Computational thinking component	Score (%) *n* = 8 groups			
			4	3	2	1
Meeting #5	Insert a template of reports that are needed in order to compile financial statements on worksheets.	Algorithmic	50%	50%		
Meeting #6	Insert formulas or codings on report templates in worksheets.	Algorithmic	50%	25%	25%	

Table 4 shows that in the first activity 50% of the college student groups executed the reporting steps in detail and in stages in worksheets and 50% of the student group executed the reporting steps in detail but not gradually into the worksheets. While in the second activity, 50% of the student group executed formulas or coding into report templates in worksheets in detail and sequentially; 25% of the student groups executed formulas or coding into report templates in worksheets in detail but not sequentially; and 25% of college student groups executed formulas or coding into report templates in worksheets were without details or on sequence. These results showed that half of the eight college student groups had used algorithmic skills in solving problems very well. Excellent algorithmic prowess will have an impact on a systematic and logical mindset.

4.3. Evaluating spreadsheets design

The evaluation of the spreadsheets design stage includes two activities that allow the development of computational thinking, namely testing and re-fining. Computational thinking that can be developed through this stage is an algorithmic component (Table 5).

Table 5. Evaluation of computational thinking in evaluating spreadsheets design

Meeting	College students' activities	Computational thinking component	Score (%) *n* = 8 groups			
			4	3	2	1
Meeting #7	Test the design that has been created and re-refine the design if the design is not already working properly.	Algorithmic	50%	50%	-	-

Table 5 shows that 50% of the student groups produce structured, automated and accurate designs and 50% of the student groups produce structured and automated but inaccurate designs. These results show that most student groups are able to test and re-refine their spreadsheets design so that it becomes a structured, automated and accurate design. College students tested whether the resulting spreadsheets design result was well automated. If the test results showed that the spreadsheets design is not working properly, then students were asked to review whether the formulas or coding in the spreadsheets design are in accordance with the planned steps. In these activities, college students are involved in the use of algorithmic thinking skills, because they have to review the steps in planning and the steps executed in worksheets.

4.4. Sharing spreadsheets design

The sharing spreadsheets design stage includes two activities that allow the development of computational thinking, namely giving criticism and suggestions. The development of computational thinking on the abstraction component can be achieved when college students provide criticism of the spreadsheet design of other groups. The components through this stage are components of abstraction and generalisation (Table 6).

Table 6. Evaluation of computational thinking in sharing spreadsheets design

Meeting	College students' activities	Computational thinking component	Score (%) *n* = 38 college students			
			4	3	2	1
Meeting #8	Presenting design results, giving criticism and suggestions to other groups' designs.	Abstraction and generalisation	26.3%	73.7%	-	-

Table 6 shows that 26.3% of students or only 10 students gave critics and suggestions to other groups' spreadsheets design, and 73.7% gave criticism only or suggestions only to other groups' spreadsheets design. These results show that spreadsheets design-sharing activities are less able to develop abstractions and generalisations. This is natural because students only have a limited time to pay attention to the designs of other groups so most of them can only give criticism or suggestions.

In addition to the analysis of the artefacts, researchers also used classroom observations to determine the extent of the development of computational thinking through the learning environment of constructionism. Students' discussion with the group does not always take place at night, they sometimes take advantage of the morning, afternoon, evening or evening. This shows that the time for discussions is flexible and the frequency of discussions outside of lecture hours is relatively frequent because social media access can be done anytime and anywhere. Behavioural and emotional engagement is very apparent from their high curiosity to learn visual basic for application (VBA) because the resulting design is more attractive than the previous design. In this intervention, college students are better able to manage cognitive load even though the cognitive load of learning coding is relatively high for accounting students because they have never learned a programming language before. The students' social engagement is seen during the process of making videos with the team, discussing work together and completing their videos.

College students become more critical in designing spreadsheets and are able to better identify design errors or shortcomings. The proficiency in abstraction, decomposition, algorithmic, generalisation and evaluation is much better when they compare the existing designs with the design parameters of spreadsheets and the company accounting cycle. The addition of the use of VBA Excel in spreadsheets design becomes more attractive and an automated design. By adding the use of VBA design activities to be more fun, especially during the debugging process, they are very enthusiastic about completing their project task. This shows that when college students are cognitively involved with excess cognitive load, they can manage that cognitive load. In other words, students' behaviour and emotional engagement such as the spirit of constant experimentation and enthusiasm can manage the cognitive load of students in learning to code. College students' satisfaction is very visible when they succeed in creating spreadsheets design.

The maximum use of social media results in creative and attractive designs. College students' behavioural and emotional engagement is very visible when executing design plans. Curiosity and high enthusiasm to complete the design can reduce the high cognitive load in the utilisation of the VBA excel. Sharing artefacts through videos before a class meeting provides an opportunity for students to take a closer look at the shortcomings or weaknesses of designs made by other groups. College students are given the opportunity to give suggestions and criticise other groups in writing and could convey them during discussions between groups in class. In other words, by sharing artefacts, college students gain meaningful experiences due to being actively involved cognitively, emotionally and socially when criticising the designs of other groups and sharing ideas with other groups.

The implementation results show that computational thinking can be developed through the activity of discussing artefacts described in the activity of discussing the results of spreadsheets design observations, constructing spreadsheets designs collaboratively and sharing spreadsheet designs using relevant platforms. Together with an appropriate learning environment, the activity of designing spreadsheets is expected to able to develop student computational thinking. The results of the design implementation are empirical evidence that the activity of designing spreadsheets based on constructionism is able to develop and improve student computational thinking. Students gain knowledge and understanding of operationalising VBA excel through building spreadsheets design. This is in line with the opinion of Ostashewski et al. (2011) who emphasise that the application of constructionist theory in learning can be carried out through the act of constructing artefacts in the framework of a process of representation of knowledge, such as software manipulation, or real actions, such as physical manipulations necessary to create artefacts.

The findings of this study are empirical evidence that college students build their meaning and understanding of the use of excel functions and coding with VBA through constructing spreadsheets design. Students engage cognitively, emotionally and socially when constructing and sharing spreadsheets design. Willis and Tucker (2001) assert that in constructionism theory the construction of meaning occurs very well when the learner is engaged in building and sharing artefacts. Artefacts are objects for thinking together (Papert, 1980) and provide learning when others see and give criticism to the resulting artefacts (Butler & Lehay, 2021).

5. Conclusion and recommendations

The constructionism learning environment in learning accounting spreadsheets is empirically able to develop and facilitate students' computational thinking. In addition, the impact of this study can also be seen in students' engagement which includes cognitive engagement, behavioural and emotional engagement and social engagement. The students' behavioural and emotional engagements at the time of executing the design plan can foster curiosity and high enthusiasm to complete the design. This engagement can reduce the high cognitive load when college students understand programming languages when utilising VBA excel. The college students' engagement when making spreadsheet designs in a constructionism learning environment strongly supports the development of their computational thinking skills. This empirical evidence can also be a best practice in spreadsheets learning, namely how to teach good spreadsheet skills and design.

Limitation this study has not empirically investigated the long-term impact of developing computational thinking, namely critical thinking skills, creativity, problem-solving, algorithmic thinking cooperation and communication, in the field of accounting. Therefore, this study can be used as a basis for conducting a subsequent study where researchers empirically investigate the impact of the development of computational thinking. In addition, this study is limited to spreadsheets learning in developing computational thinking and thus further study on the constructionism learning environment can be implemented in other learning in the field of accounting. These constructionism-based spreadsheet design activities can be adopted and implemented by learning practitioners to improve students' quality so that they can compete in this digital era.

Acknowledgements

The authors thank the Universitas Sebelas Maret for the whole support.

References

Acevedo-Borrega, J., Valverde-Berrocoso, J., Garrido-Arroyo,M.d.C. (2022). Computational Thinking and Educational Technology: A ScopingReview of the Literature. *Educ. Sci.*, 12, 39. https://doi.org/10.3390/educsci12010039

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education. *Publications Office of the European Union*. https://doi.org/10.2791/792158

Bock, A. A., Bøgholm, T., Sestoft, P., Thomsen, B., & Thomsen, L. L. (2020). On the semantics for spreadsheets with sheet-defined functions. *Journal of Computer Languages*, 57, 100960. https://doi.org/10.1016/j.cola.2020.100960

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, 42(3). http://dx.doi.org/10.14221/ajte.2017v42n3.4

Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology*, 52(3), 1060–1077. http://dx.doi.org/10.1111/bjet.13090

Frownfelter-Lohrke, C. (2017). Teaching good Excel design and skills: A three spreadsheet assignment project. *Journal of Accounting Education*, 39, 68–83. http://dx.doi.org/10.1016/j.jaccedu.2016.12.001

Gero, A., & Levin, I. (2018) Computational thinking and constructionism: creating difference equations in spreadsheets. *International Journal of Mathematical Education in Science and Technology*. http://dx.doi.org/10.1080/0020739X.2018.1501827

Girvan, C., & Savage, T. (2019). Virtual worlds: A new environment for constructionist learning. *Computers in Human Behavior*, 99, 396–414. http://dx.doi.org/10.1016/j.chb.2019.03.017

Hay, K. E., & Barab, S. A. (2001). Constructivism in practice: A comparison and contrast of apprenticeship and constructionist learning environments. *Journal of the Learning Sciences*, 10(3), 281–322. http://dx.doi.org/10.1207/s15327809jls1003_3

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(1), 296–310. http://dx.doi.org/10.1016/j.compedu.2018.07.004

Israel-Fishelson, R., & Hershkovitz, A. (2022). Studying interrelations of computational thinking and creativity: A scoping review (2011–2020). *Computers & Education*, 176, 104353. https://doi.org/10.1016/j.compedu.2021.104353

Jones, B. F., Rasmussen, C. M., & Moffitt, M. C. (1997). Real-life problem solving: A collaborative approach to interdisciplinary learning. American Psychological Association. https://psycnet.apa.org/record/1997-97254-000

Lawson, B., Baker, K., Powell, S., & Fosterjohnson, L. (2009). A comparison of spreadsheet users with different levels of experience. *Omega*, 37(3), 579–590. http://dx.doi.org/10.1016/j.omega.2007.12.004

Niyazova, G. Z., Saparkhojayev, N.P., Bazarbaeva, A.I., Azybayev, M.A. (2022). Development of digital competence of school teachers. *World Journal on Educational Technology: Current Issues*. 14(3), 592-603. https://doi.org/10.18844/wjet.v14i3.7196

Ostashewski, N. M., Reid, D., & Moisey, S. (2011). Applying constructionist principles to online teacher professional development. The International Review of Research in Open and Distributed Learning, 12(6), 143–156.

Papert, S. (1980). Mindstorms, children, computers, and powerful ideas. Basic Books.

Resnick, M. (1996). Distributed constructionism. In D. C. Edelson, & E. A. Domeshek (Eds.), International Conference on the Learning Sciences, 1996 (pp. 280–284). Association for the Advancement of Computing in Education (AACE).

Schneider, K. N., Becker, L. L., & Berg, G. G. (2017). Beyond the mechanics of spreadsheets: using design instruction to address spreadsheet errors. *Accounting Education*, 26(2), 127–143. http://dx.doi.org/10.1080/09639284.2016.127491

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. https://eprints.soton.ac.uk/356481/

Sudjimat, D. A., Nyoto, A., & Romlie, M. (2021). Implementation of project-based learning model and workforce character development for the 21st century in vocational high school. *International Journal of Instruction*, 14(1), 181–198. https://doi.org/10.29333/iji.2021.14111a

Tsai, M. -J., Liang, J. -C., & Hsu, C. -Y. (2020). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, 59(4), 579–602. https://doi.org/10.1177/0735633120972356

Willis, E. M., & Tucker, G. R. (2001). Using constructionism to teach constructivism: Modeling hands-on technology integration in a preservice teacher technology course. *Journal of Computing in Teacher Education*, 17(2), 4–7.

Yeh, K., Xie, Y., & Ke, F. (2011). Teaching computational thinking to non-computing majors using spreadsheet functions. Frontiers in Education Conference (FIE). https://doi.org/10.1109/fie.2011.6142980