

Threshold concepts in Computer Science teaching



Teaching (Today for) Tomorrow: Bridging the Gap between the Classroom and Reality

3rd International Scientific and Art Conference
Faculty of Teacher Education, University of Zagreb in
cooperation with the Croatian Academy of Sciences and
Arts

Gabrijela Jakovac, Martina Holenko Dlab

*University of Rijeka, Faculty of Informatics and Digital Technology
gabrijela.jakovac@student.uniri.hr*

**Section - Education for digital
transformation**

Paper number: 46

**Category: Original scientific
paper**

Abstract

Fundamental concepts underlie every scientific field. Among them, there are concepts that represent a turning point in the understanding of the field and whose understanding is a significant challenge for students. Such concepts are called threshold concepts. The aim of this paper is to provide an overview of the characteristics of threshold concepts that distinguish them from fundamental concepts, to identify threshold concepts in the field of computer science, and to emphasize the need for selecting appropriate teaching strategies and approaches for teaching threshold concepts using digital technology.

In addition to the list of threshold concepts in computer science derived from the literature review, a list of threshold concepts derived from research with computer science teachers is presented. The nominal group technique, which provides a structured approach to idea exchange within the group, was used to identify threshold concepts. Participants (N=53) first proposed the threshold concepts individually by writing explanations and then presented them to the group. The group discussed and voted to reach a consensus. In identifying threshold concepts, the focus was on recognizing transformative and integrative features to identify concepts whose understanding triggers a significant shift in the understanding of the subject area and makes connections that were previously hidden.

Identifying threshold concepts can help guide learning and teaching. With a better understanding of the difficulties students face, teachers can provide personalized support to help students master these concepts using technology. Further research will focus on analyzing the possibilities of applying approaches for teaching threshold concepts, especially game-based learning approaches.

Key words:

computer science, personalization, STEM, teaching, threshold concept.

Introduction

In today's digital society, in which we have grown up and are active members, it is clear that the integration of information technology within the educational system is becoming increasingly significant (Bognar, 2016). In the context of computer science, teachers face the dual challenge of

teaching foundational knowledge and enabling students to overcome key barriers to understanding. These obstacles include threshold concepts— core ideas that represent a transformative points in learning. These concepts are not only fundamental, but also serve as a gateway to deeper understanding that often requires a significant shift in perspective to master.

Meyer and Land (2003) have identified the threshold concept in the field of education as a set of ideas that, once understood, become transformative but are initially challenging and unfamiliar. Regardless of whether we adopt a constructivist approach or another learning theory, threshold concepts represent points at which students are likely to encounter learning difficulties. To further define threshold concepts, Meyer and Land state that they are integrative, as they show previously unknown ways of linking ideas; irreversible, as the new way of thinking becomes part of the learner once they have truly understood it; and boundary markers, as they define the boundaries of part (or all) of a set of ideas. An entire subject area may have its boundary marked by a single threshold concept, mastery of which indicates competence in that area (Meyer & Land, 2003).

Dr. Tucker highlights several characteristics of threshold concepts and emphasizes five main characteristics: transformative, irreversible, integrative, troublesome, and bounded (SJSU School of Information, 2013).

Research on threshold concepts in computer science has highlighted certain concepts as transformative and challenging for students as they often require significant cognitive change. Among these concepts, object-oriented programming stands out due to its complexity and potential to enable deeper understanding and application in different areas of computer science (Boustedt et al., 2007). However, the concepts taught in primary school have not been the focus of such research, leaving a gap in the understanding of threshold concepts that younger students should overcome.

This paper examines the threshold concepts in computer science education, with a focus on primary school. The research aims to identify concepts that are particularly challenging for students and to distinguish those that can be considered threshold concepts based on their characteristics. Using the nominal group technique, primary school teachers were involved in a structured research process and threshold concepts were proposed on the basis of shared insights. By addressing this topic, this research aims to improve teaching practices and support learning and teaching, as by understanding these concepts, teachers can better address the obstacles students face and guide them towards understanding the key concepts.

Related work

Characteristics of threshold concepts in education

In this section, the following characteristics of threshold concepts are described according to Meyer & Land (2003): *transformativeness*, *irreversibility*, *integrativeness*, *troublesomeness*, and *boundedness*.

Transformative is described and associated with events that leave a lasting impression and are unforgettable, such as passing a driving test. The transformation of attitudes, values or understanding often represents a decisive point in our lives. This change not only shapes our identity, but also has a profound impact on our daily lives. Through this development, the new understanding gradually integrates into our biography and becomes an inseparable part of who we are. This process does not happen immediately, but unfolds gradually and permeates all aspects of our lives. It is not just a matter of adopting a new attitude or a new value. Rather, this transformation becomes part of our inner being and shapes the way we perceive the world around

us. Through this integration, the new understanding becomes a fundamental element of our identity and influences our thoughts, feelings and actions. This change does not occur in isolation, but has a profound impact on our relationships, our work and our life choices. The importance of such transformations lies in their ability to promote growth and development as individuals. They encourage us to look at things from different angles, which gives us a broader view of life. Furthermore, these changes often coincide with personal growth and make us stronger and more resilient to life's challenges.

Irreversibility, in the context of knowledge, stands for a profound level of learning, where what we have mastered becomes an integral part of our intellectual repertoire. It is like riding a bike or swimming, where a learnt skill becomes inherent and indelible. Through the process of irreversibility, knowledge becomes imprinted in our memory in a way that resists forgetting, even in challenging situations. This phenomenon can be compared to riding a bike. As soon as we master a technique, it becomes part of our muscle memory. No matter how long we have not ridden a bike, when we do it again, the process naturally emerges from our subconscious. Similarly, irreversibility in learning means that once acquired, knowledge becomes a permanent skill that is activated regardless of a prolonged period of disuse.

Integrative learning means that what was previously hidden or not fully understood is made accessible in its context. This quality of learning has the power to link separate concepts together so that they are brought together into a holistic understanding. Ideas that were once separate are now connected, creating a broader understanding that enriches individual perspectives. This integration process can be likened to putting together pieces of a jigsaw puzzle. Each individual piece represents a particular concept, and through integration, these individual pieces become an integral part of a larger, complete landing. Furthermore, integrativeness is not just about putting different concepts together, it goes a step further by creating an expanded understanding that enriches our perception of the world around us. This dimension of integrativeness significantly influences the development of individual understanding. Ideas that have previously isolated now become part of a wider network of connections, leading to a richer and deeper experience of knowledge.

Troublesomeness, in learning can be associated and described with certain concepts that may seem counter-intuitive or unpleasant. However, it is essential to face these challenges in order to understand them. Often these concepts are associated with situations that cause discomfort or are counter-intuitive, and this discomfort may stem from misconceptions. Especially when solving problems in physics, beginners are often confronted with various misconceptions and contradictions. However, when they dedicate themselves to solving these challenges, they not only overcome their preconceptions but also reach a new level of understanding. Wrestling with counterintuitive ideas becomes a path to deeper understanding, and this process allows individuals to reach new heights in their learning eliminating underestimation in the process.

Boundedness implies the presence of definitive boundaries. These boundaries serve as transitions between different conceptual areas, defining boundaries to other thresholds and introducing us to new areas of understanding. In a particular subject area, specialized terminology takes on a new meaning defined precisely by these boundaries. Boundedness implies not only the presence of endpoints, but also the possibility of exploring and expanding these boundaries to deepen our understanding and discover new meanings that emerge within these defined boundaries (Mayer and Land, 2003), (SJSU School of Information, 2013).

Threshold concepts in computer science

The research on threshold concepts in computer science conducted by Boustedt et al. (2007) focused on identifying terms that could correspond to threshold concepts and validating them with students, followed by checking whether the criteria for threshold concepts are met. At the Conference on Innovations in Computer Science Education in June 2005, 33 computer science experts from nine countries were surveyed to select terms that met the criteria for threshold concepts. In November 2005, a similar study was conducted at a conference on computer science in Finland, the results of which focused on the „hard to learn” aspects of threshold concepts (McCartney and Sanders, 2005). Subsequent studies at different universities in several countries showed that students identified „control structures”, „sequential thinking”, „parameters”, „objects” and „memory models” as threshold concepts. Object-oriented technologies and pointers were selected for in-depth analysis as they fully met all the criteria for threshold concepts (Boustedt et al., 2007).

Object-oriented programming (OOP) is based on the idea that a program consists of objects that represent interconnected parts of a solution, in contrast to the classic procedural model, which views a program as a sequence of instructions. OOP enables more efficient code organization and simplifies the maintenance and scaling of large programs. It is used in languages such as Java and Python, while the procedural model is suitable for languages such as C and Pascal (Jovanović, 2012). Although OOP offers numerous advantages, students often report difficulties in learning it, especially with basic concepts such as classes and objects. Research among first-year university students shows that many have experienced OOP as a challenge that requires rethinking. Nevertheless, most students emphasize that mastering OOP has helped them to understand more complex programming concepts and enabled them to transfer the skills learned to other areas, such as software engineering, demonstrating the transformative nature of OOP. Such experiences show that although the learning process is long and complex, it provides long-term benefits and enables students to apply their knowledge in different contexts (Boustedt et al., 2007).

Pointers have been identified as a threshold concept in computer science because their understanding is often challenging for students, especially when they are used as parameters in programs. Students reported difficulty connecting abstract theory to the practical application of pointers. One student described having difficulty understanding pointers until he realized that they simply represent a specific memory location, which made the concept clearer. Having mastered pointers, students began to apply this knowledge in broader contexts such as hardware and operating systems by using pointers in practical work, for example in assembly language. Understanding pointers enabled them to apply objects and references more successfully in programming, leading to greater confidence in tackling more complex computing problems (Boustedt et al., 2007).

Recent research by Kallia and Sentance (2020) provides additional insights in the context of functions. Their study emphasises the transformative and integrative nature of concepts such as „parameters”, „parameter passing”, „return values”. These concepts were found to be challenging for secondary school students and their mastery facilitates deeper understanding and integration of knowledge across programming topics. The authors identified „procedural decomposition” as a potential threshold skill requiring extensive practice to bridge theoretical understanding and application.

The study conducted by McSkimming, Mackay & Decker (2023) identified two threshold concepts for intermediate computer science students: „algorithmic runtime” and „memory management”, emphasizing the challenges and transformative understanding associated with

these concepts. The research by Govender & Olugbara (2022) reflected on threshold concepts for developing programming skills in first-year information technology students, providing insights into supporting teaching strategies and identified procedures and functions and programming constructs (such as selection, iteration, and variable manipulation) as critical threshold concepts for teaching computer programming to first-year IT students.

These studies highlight the ongoing interest in understanding and addressing threshold concepts to enhance computer science education.

Methodology

The main aim of this paper is to propose threshold concepts for computer science taught in primary school. The basic methodology of the research was the nominal group technique (NGT), which is suitable for gaining a deeper insight into the participants' perception and understanding of threshold concepts. The nominal group technique (NGT) was found to be a very effective method for promoting critical thinking through discussions that involve a small number of participants, provide clear and focused instructions and allow for constructive feedback. NGT fulfils all these criteria while ensuring the full participation of all group members, which is especially useful in an educational context.

NGT is widely used in various disciplines, e.g. medicine, information technology, politics, management and education, where it serves as a method for evaluated discussions (Macphail, 2001). In education, NGT is used in the design and evaluation of curricula and as a pedagogical method that encourages active participation (Chapple & Murphy, 1996). Studies have shown that NGT increases participant's productivity and problem-solving skills through structured discussions (Madar, 1982).

Procedure

The application process of NGT consisted of six steps, beginning with an oral *presentation* that covered the definitions and examples provided. A presentation was prepared, which included precise definitions of threshold concepts, along with examples of threshold concepts with explanations according to their characteristics. For demonstrating threshold concepts, „looping“ and subprograms were selected as examples that fully align with the threshold concept. Simultaneously, as an example of a demanding concept, recursion was analyzed and identified as a concept that requires additional understanding but is not a necessary prerequisite for further learning in programming.

The second step, *silent idea generation*, required participants to individually reflect on and write down their thoughts regarding threshold concepts. They completed a questionnaire that required them to try to identify concepts and answer whether these concepts are fundamental, demanding, and whether they meet the characteristics of threshold concepts.

The third step, *idea discussion*, took place after individual reflections. Participants were divided into smaller groups with 4 to 7 participants to further deepen their understanding and discuss the proposed threshold concepts. Through constructive discussion, participants contributed to a deeper understanding of the concepts and their application in computer science education. In this group decision-making phase, each group selected one or more concepts that they believed met all the characteristics of threshold concepts.

To obtain clear results and rank the proposed threshold concepts, *voting* was conducted via the online platform Padlet. After the voting and reviewing of the results, *conclusions* were drawn, and potential threshold concepts were identified. Lastly, the *report* with a summary of the procedure, decisions, and final results was written.

Participants

The participants of the research were primary school teachers of computer science (N=53) who participated in the activities organized by the Professional Council of Computer Science Teachers of Primorje-Gorski Kotar County at Vežica Primary School in Rijeka in February 2024.

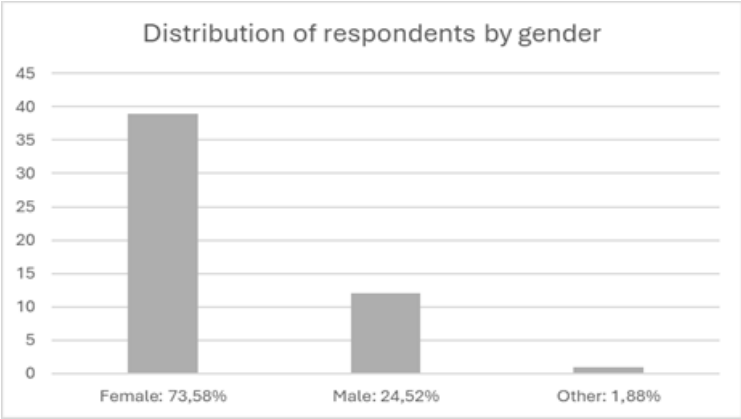
In addition to the questions on threshold concepts, participants were asked to provide demographic data, including their educational profile, professional experience and the subjects they teach. Table 1 shows the age and gender of the participants. The participants in the survey have a wide age range, from 24 to 63 years, with an average age of 39 years. This age diversity suggests that participants are at different stages in their careers, which could influence their responses given their varied teaching experience.

Table 1
Age of the participants

Age	Number of participants	F	M	Others
24-34	21	15	6	0
35-44	10	7	3	0
45-54	19	15	3	1
55+	3	2	1	0

The majority of participants were women, a smaller number were men (Chart 1). One respondent did not specify their gender.

Chart 1
Distribution of participants by gender



All participants have a high level of education and work as computer science teachers at primary schools (Table 2). The average length of time spent in the teaching profession is around 10 years. A total of 33 participants have between 0 and 9 years of professional experience, 7 participants have between 10 and 19 years of professional experience, 11 participants have between 20 and 29 years of professional experience, and 2 participants have more than 30 years of professional experience.

Table 2

Participants' work experience

Work experience	Number of participants	F	M	Others
0-9	33	23	10	0
10-19	7	6	1	0
20-29	11	9	1	1
30+	2	1	1	0

Participants vary in the combination of subjects they teach, including computer science, mathematics, English, physics and technical education what can be seen in Table 3.

Table 3

Number of participants by job position

Job Position	Number of Participants
Teacher of Computer Science	39
Teacher of Computer Science and Mathematics	11
Teacher of Computer Science and Technical Culture	2
Teacher of Computer Science and English Language	1

Results

The results collected through the survey provide insight into opinions of computer science teachers regarding the threshold concepts.

Within the stage of *silent idea generation*, the participants highlighted several key fundamental concepts in computer science education, including „Branching“, „Functions“, „HTML“, „File Storage“, „Addressing“ and „Logical Conditions“, which were recognized as essential for understanding computer science. „Branching Algorithm“ was marked as a fundamental concept by 15 participants, while „Functions“ and „Logical Conditions“ received 13 votes. „HTML“ and „File Storage“ were also identified as fundamental by 11 participants, and „Addressing“ was also ranked highly.

Challenging concepts that represent learning obstacles include „Branching“, „Database Relations“ and „File Storage“, which require additional attention and resources for adequate

understanding.

The interconnection of concepts was also important, and the participants recognized that „Branching“, „Functions“, „Logical Conditions“, „HTML“ and „File Storage“ are connected, meaning that mastering these concepts facilitates understanding of other aspects of computer science.

The transformative nature of concepts, which indicates their potential to lead to deeper understanding, was also highlighted for „Branching“, „Functions“, „HTML“ and „Logical Conditions“, while the irreversibility of these concepts was considered high, meaning that students rarely forget these terms once they have mastered them.

Concepts such as „Database Relations“ and „Branching“ were marked as bounded, meaning that their understanding may vary among students, requiring continuous repetition.

Some concepts, such as „Attributes“, „Database“, „File“, „Elements in Canva“, „Internet“, „Output Devices“, „Copying“ and „References in Word“, were not recognized as fundamental or challenging by most participants, indicating their lower importance in basic computer science education.

During the *idea discussion* phase, the participants concluded that some concepts, such as „Branching“ and „Logical Conditions“, could be combined because they are interconnected, and that „Data Organization“ and „File Storage“ could also be merged, as data organization is key to understanding the process of file storage. After the *discussion* phase, we move on to the *voting* phase.

In order to obtain clear results and ranking of the proposed threshold concepts, a voting was conducted through the online platform Padlet. In the following table, the proposed threshold concepts and the voting results for each proposed concept can be seen. Because some teachers were undecided on certain concepts, the total number of votes differs per concept and is less than 53.

Table 4

Proposed threshold concepts

PROPOSED CONCEPTS	N	YES	NO
Logical conditions	43	42	1
Cell addressing (in spreadsheet)	42	40	2
Variables	43	39	4
Data organization (storage)	45	39	6
Flowchart	41	35	6
Relations (databases)	42	31	11
HTML and similar languages	39	29	10
User account	42	24	18
Personal data protection	42	15	27

By analyzing the voting results, the list of threshold concepts was proposed based on input from participants, highlighting those that clearly received higher support as threshold concepts in computer science education. „Logical conditions“ received an exceptionally high number of votes, with 42 votes „YES“ and only one vote „NO“. „Cell addressing (in spreadsheet)“ also received a large number of votes, with 40 votes „YES“ and only two votes „NO“. „Flowchart“ received 35

votes „YES” and only 6 votes „NO.” „Data organization(storage)” received 39 votes „YES” and 6 votes „NO”. „Variables”, although not receiving unanimous support, can be considered a threshold concept given the predominant support.

Discussion

The results of the survey provide several important insights into primary school IT teachers' views on threshold concepts. Concepts such as „logical conditions”, „cell addressing”, „variables”, and „data organization” received strong support, indicating their importance for computer science teaching. The high consensus on „logical conditions” (42 out of 43 participants who voted for this concept) may reflect its fundamental role in programming, particularly in helping students understand decision-making in algorithms. This strong support suggests that teachers view it as a prerequisite for grasping more advanced programming concepts.

In contrast, the slightly lower support for the concept of „flowchart” (35 votes) may indicate variability in how teachers integrate visual algorithm representation into their teaching practices. While flowcharts are valuable for structuring and planning code, some teachers may rely on alternative methods, such as pseudocode, to introduce these skills.

The strong support for „cell addressing” (40 votes) highlights its relevance not only for spreadsheet data management but also for broader applications like data analysis. This result underscores the increasing emphasis on data literacy in primary education, aligning with the growing importance of data science skills. However, it also raises questions about whether teachers feel adequately equipped to teach these skills effectively.

These findings suggest that while core programming concepts such as „logical conditions” and „variables” are well-established in the curriculum, there is room to explore how concepts like „data organization” and „flowcharts” can be better integrated. Future research might investigate how these concepts are taught in practice and the challenges teachers face in developing student's understanding of them.

Conclusions

The topic of threshold concepts represents not only a challenge, but also an opportunity for innovation in computer science teaching. Clearly defined threshold concepts provide a foundation for high-quality education and promote the development of the skills required in the digital age.

Methodologies like the nominal group technique allow structured dialogue among teachers, creating a space for exchanging experiences and finding common solutions to teaching challenges. Based on the research conducted using the nominal group technique with primary school teachers the following threshold concepts in computer science education have been proposed: logical conditions, cell addressing (in spreadsheet), flowcharts, data organization, and variables. These concepts have been highlighted as transformative points that enable deeper understanding of the subject matter and progress for students.

In the future work, these proposals will be validated through further research involving subject-matter experts, who will contribute to developing a solid rationale confirming that these concepts have the defining characteristics of threshold concepts. The integration of educational games has considerable potential to support students in mastering complex topics while increasing their engagement and motivation in the learning process. Following this validation, game-based

learning activities will therefore be designed to facilitate student's acquisition of these concepts.

Acknowledgment

The research has been funded by the Erasmus+ Programme of the European Union, KA220-SCH - Cooperation partnerships in school education, under the project "Science&Math educational games from preschool to university" (023- 1-HR01-KA220-SCH-000165485).

Literature

- Bognar, B. (2016). Teorijska polazišta e-učenja. Croatian Journal of Education: Hrvatski časopis za odgoj i obrazovanje, 18(1), 225-256. Available online: <https://hrcak.srce.hr/file/229424>, accessed 28.12.2023.
- Boustedt, J., Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2007). Threshold concepts in computer science: do they exist and are they useful? ACM Sigcse Bulletin, 39(1), 504-508.
- Breidenbach, D., Dubinsky, E., Hawks, J., & Nichols, D. (1992). Development of the process conception of function. Educational studies in mathematics, 23(3), 247-285.
- CELatElon, Center for Engaged Learning. (2013, September 18). Decoding the disciplines and threshold concepts. Accessed December 10, 2023, Available online: https://youtu.be/Wqe_kKFoOq4.
- Falkner, N. J. G., Vivian, R. J., & Falkner, K. E. (2013). Computer Science Education: The First Threshold Concept. 2013 Learning and Teaching in Computing and Engineering. Macau: Macao, pp. 39-46. Available online: <https://doi.org/10.1109/LaTiCE.2013.32>.
- Gruić, I. (2005). *Entering the story labyrinth. An Investigation of the Effects of Placing the Participants in Different Actantial Positions within the Dramatic World Created by a Theatre in Education Programme*. [Doctoral disertation or Master's thesis]. Birmingham: Faculty of Education University of Central England in Birmingham.
- Hoić-Božić, N., & Holenko Dlab, M. (2021). Uvod u e-učenje: obrazovni izazovi digitalnog doba. Rijeka: Sveučilište u Rijeci, Odjel za informatiku.
- Jovanović, N. (2012). Objektno orijentisano programiranje. Kruševac: SIGRAF, Visoka poslovna škola.
- Kallia, M., & Sentance, S. (2021). Threshold concepts, conceptions and skills: Teachers' experiences with students' engagement in functions. *Journal of Computer Assisted Learning*, 37(2), 411-428.
- Klafki, W., Schulz, W., Cube, F., Moller, C., Winkel, R., & Blankertz, H. (1992). Didaktičke teorije. Zagreb: Educa.
- Kyriacou, C. (1995). Nastavna umijeća: Metodički pristup za uspješno podučavanje i učenje. Zagreb: Educa.
- Macphail, A. (2001). Nominal group technique: A useful method for working with young people. British Educational Research Journal, 27(2), 161-170. Accessed May 1, 2024, Available online: <https://doi.org/10.1080/01411920120037117>.
- Madar, D. (1982). Using nominal group technique to foster productive behavior in group discussions. Teaching Political Science, 9(4), 185-189. Accessed May 1, 2024, Available online: <https://doi.org/10.1080/00922013.1982.9942918>.

- McCartney, R., & Sanders, K. (2005). What are the threshold concepts in computer science? U: Proceedings of the Koli Calling 2005 Conference on Computer Science Education, T. Salakoski & T. Mäntylä (ur.), str. 185.
- McSkimming, B. M., Mackay, S., & Decker, A. (2023, October). Identification of Threshold Concepts for Intermediate Computer Science Students. In 2023 IEEE Frontiers in Education Conference (FIE) (pp. 1-5). IEEE.
- Meyer, J. H. F., & Land, R. (2003). Threshold Concepts and Troublesome Knowledge Linkages to Ways of Thinking and Practising. U: Improving Student Learning Ten Years On. Rust, C. (ur.). Oxford.
- Ministarstvo znanosti i obrazovanja [MZO]. (2018). Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije. Accessed April 12, 2024, Available online: https://skolazazivot.hr/wp-content/uploads/2020/06/INF_kurikulum.pdf.
- Ministarstvo znanosti i obrazovanja [MZO]. (2019). Kurikulum međupredmetne teme Učiti kako učiti za osnovne i srednje škole. Accessed April 12, 2024, Available online: <https://tinyurl.com/45rv8bu4>.
- Olugbara, Thamotharan P. Govender¹ Oludayo O. "Reflection on Threshold Concepts for Developing Computer Programming Skills of First-year Information Technology Students."
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM, 52(11), 60-67.
- SJSU School of Information. (2023). The Expert Searcher and Threshold Concepts. Accessed December 10, 2023, Available online: <https://youtu.be/4l1Ue0vpcMw>.
- Vinner, S., & Dreyfus, T. (1989). Images and definitions for the concept of function. Journal for research in mathematics education, 20(4), 356-366.

Revizija #3

Stvoreno 20 svibnja 2025 18:41:37 od Martina Gajšek

Ažurirano 21 svibnja 2025 07:37:44 od Martina Gajšek