The Pedagogical Alignment of Computational Thinking to Architecture Education for the 21st Century Learners

Elif Öksüz Uncu[1], Gülen Çağdaş[2]

ORCID NO: 0000-0002-7807-171X[1], 0000-0001-8853-4207[2]

[1]Kahramanmaraş İstiklal University, Faculty of Engineering, Architecture and Design, Architecture Department, Kahramanmaraş, Turkey
[2] Istanbul Technical University, Graduate School, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

For the 21st century learners, Millennials and Gen-Z students, the concept of Computational Thinking (CT) has been inclusively affirmed in higher education with different teaching methods and strategies. However, it has been almost a decade that Generation Z students form the main bulk of students in classrooms. And their distinct characteristics from the Millennials have necessitated rethinking educational practices, pedagogies, and teaching approach to provide an optimal and holistic learning environment that meets their learning needs. In this regard, by scrutinizing the contemporary approach to the concept of Computational Thinking, this article discusses the pedagogical alignment of CT in architecture education by addressing its cognitive contributions as a mental tool for the 21st century learners. It highlights the challenges of teaching computational thinking within the current pedagogical framework in architecture education by regarding the learning preferences and attributes of Generation-Z.

Keywords: Architecture Pedagogy, Cognitive Skills, Computational Thinking, Generation-Z. Millennials.

Theoretic Article

Received: 20.07.2022
Accepted: 06.08.2022

Corresponding Author:
elifbelkis.oksuz@istiklal.edu.tr

Uncu Öksüz, E. & Çağdaş G. (2022). The pedagogical alignment of computational thinking to architecture education for the 21st century learners. *JCoDe: Journal of Computational Design*, 3(2), 159-172. https://doi.org/10.53710/jcode.1146123

Yüzyıl Öğrencileri için Hesaplamalı Düşünmenin Mimarlık Eğitimine Pedagojik Uyumu

Elif Öksüz Uncu[1], Gülen Çağdaş[2]

ORCID NO: 0000-0002-7807-171X[1], 0000-0001-8853-4207[2]

[1]Kahramanmaraş İstiklal Üniversitesi, Mühendislik, Mimarlık ve Tasarım Fakültesi, Mimarlık Bölümü, Kahramanmaraş, Türkiye
[2] İstanbul Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

21. yüzyıl öğrencileri (Y ve Z Kuşakları) için, Hesaplamalı Düşünme (HD) kavramı, farklı öğretim yöntemleri ve stratejileri ile yükseköğretimde kapsayıcı bir şekilde teşvik edilmektedir. Ancak bu süreçte geliştirilen ve uygulanan stratejilerin iki farklı kuşağı kapsaması eğitim pedagojisi açısından dikkate değerdir. Nitekim yaklaşık on yıldır, sınıflardaki öğrencilerin büyük çoğunluğunu Z Kuşağı oluşturmaktadır. Y Kuşağı'nın aksine, Z Kuşağının günlük yaşamlarının bir parçası olarak teknolojiye bağlılığı onları 'gerçek dijital yerliler' olarak tanımlamaktadır. Bu kuşağın Y Kuşağı'ndan farklı özellikleri, öğrenme ihtiyaçlarını karşılayan optimal ve bütünsel bir öğrenme ortamı sağlamak için eğitim uygulamalarının, pedagojilerinin ve öğretim yaklaşımının yeniden düşünülmesini gerektirmiştir. Bu kuşak öğrencileri beceri odaklıdır, ancak aynı zamanda tekrarlarla güncellenebilen ve geliştirilebilen yaşam boyu öğrenme becerilerini de benimserler. Odaklanmak, kendi hızlarını belirlemek ve öğrenmelerini anlamlandırmak için bireyselleştirilmiş öğrenmeyi, ilgi çekici ve görsel öğrenme ortamlarını tercih ederler. Ama aynı zamanda, bir öz-değerlendirmeye ve öğrenirken anında ve bireysel geri bildirime ihtiyaç duyarlar. Bu bağlamda, bu makale Hesaplamalı Düşünme kavramına dair çağdaş yaklaşımı inceleyerek, 21. yüzyıl öğrencileri için zihinsel bir araç olarak bilişsel katkılarını ele almakta; Z kuşağına yönelik mimarlık eğitiminde HD'nin pedagojik uyumunu tartışmaktadır. Mimarlık eğitiminde mevcut pedagojik çerçeve içinde hesaplamalı düşünmeyi öğretmenin zorluklarını, Z Kuşağının öğrenme tercihlerine ve niteliklerine yer vermektedir.

21. yüzyılda mimarlık öğrencileri, erken dönemlerden başlayarak eğitimleri boyunca dijital tasarım araçları ve etkileşimli araçlarla çalışmaktadır. Öğrencilerden tasarımlarında hesaplama açısından pahalı çözümlerden kaçınarak dijital teknolojilerle yaratıcı, etkili ve verimli çalışmaları beklenmektedir. Ve bu tür bir katılım, öğrenciler ve eğitimciler için yeni bir dizi iş akışına uyum sağlamayı beraberinde gerektirmektedir. Başka bir deyişle, tasarım teknolojilerinin farklı şekillerde aktif olarak kullanılması, hem öğrencilerin hem de tasarım eğitimcilerinin bilişimsel düşünme konusunda yetkin olmasını gerektirmektedir. Öte yandan çoğu mimarlık okulunda, öğrencilerin dijital teknoloji yetkinliği, bilgisayar okuryazarlığı veya bilgisayar destekli tasarım teknolojilerindeki teknik becerileri ile bağlantılıdır; ve genellikle bu beceriler 'Bilgisayar Destekli Tasarım' dersleri kapsamında ele alınmaktadır. Mimarlık okullarında teknolojik donanımın yetersiz olması halinde eğitimciler, kalabalık sınıflarda standartlaştırılmış ders alıştırmalarını tercih etmekte ve öğrencilerinin bireysel öğrenme ihtiyaçlarını göz ardı etmektedir. Bu nedenle, öğrenciler becerileri için anında geri bildirim alabilseler de eğitimciler, öğrencilerin bilişsel becerilerini bireysel gereksinimlerine göre değerlendirmekte zorlanmaktadır.

Anahtar Kelimeler: Bilişsel Beceri, Hesaplamalı Düşünme, Mimarlık Pedagojisi, Y-Kuşağı, Z-Kuşağı.

Teorik Makale

Teslim Tarihi: 20.07.2022
Kabul Tarihi: 06.08.2022

Sorumlu Yazar:
elifbelkis.oksuz@istiklal.edu.tr

Uncu Öksüz, E. & Çağdaş G. (2022). Yüzyıl öğrencileri için hesaplamalı düşünmenin mimarlık eğitimine pedagojik uyumu. *JCoDe: Journal of Computational Design*, 3(2), 159-172.
https://doi.org/10.53710/jcode.1146123

1. INTRODUCTION

The cognitive aspects of technology use in our daily lives have set new foundation skills for humans and machines. Our daily activities involve offloading information and computational processes to external tools, and reloading the information and computational outcomes back to our internal processes (Cecutti et al., 2021). And this continuous information flow between the human and computing environment requires us to acknowledge different modes of thinking. While the technology is expected to be *'brain friendly'* to complement the cognitive processes of the human user (Dror, 2011), the human mental processes are expected to become similar to the series of internal states produced as if a computer carries out a program; so one should be able to devise computer programs that mimic or simulate human thinking (Weisberg & Reeves, 2013). In this regard, we are currently facing a paradigm in which Computational Thinking (CT) is reintroduced as a mental tool to keep this continuous information flow.

Although the concept of CT was formerly introduced through the cognitive studies in computer science, the interests of outer disciplines to this mode of thinking seems relatively new. From the interdisciplinary use of digital technologies in different forms to the development of new computation models, all have caused changes in the conceptual framework of CT and its pedagogical alignment to different fields of education. And thanks to Jeannette Wing's groundbreaking approach (Wing, 2006), CT is now promoted as a combination of certain soft skills and hard skills which can be contextualized towards disciplinary frameworks.

With her intriguing article, "Computational Thinking" in the Association of Computing Machinery, Wing (2006) reintroduced the concept of computational thinking "as a mental tool that helps solving problems and understanding human behavior by drawing on the concepts fundamental to computer science" (Wing, 2008). Since then, this mental tool has been inclusively endorsed as a 21st century skill in higher education for the 21st century learners (Millennials and Generation Z). Nonetheless, it has been almost a decade that the Generation Z students have replaced the Millennials and bulked the classrooms in higher education. And their distinct characteristics from the Millennials have necessitated the need for educators to alter educational practices, pedagogies and teaching approach to provide an optimal and holistic learning environment that meets their learning needs (Shorey et al. 2021). In this regard, by scrutinizing the contemporary approach to the concept of Computational Thinking, this article discusses the pedagogical alignment of CT in architecture education by addressing its cognitive contributions as a mental tool for the 21st century learners. Later, it highlights the challenges of teaching computational thinking within the current pedagogical framework in architecture education by regarding the learning preferences and attributes of Generation-Z.

2. RETHINKING COMPUTATIONAL THINKING FOR THE 21ST CENTURY LEARNERS

Computational thinking might have gained its popularity within the last ten years, but the outset of this concept extends to Herbert Simon and Alan Newell's *'general problem-solving model'* (Simon & Newell, 1971). In this model, the analogy of the mind as a computer seems at the functional level, or at the level of the software (Weisberg & Reeves, 2013). In other words, this model introduces Computational Thinking as a machinery skill for computer scientists, leaning on the hard skills of computing, such as numeric computation and procedural thinking. Hence, the former reflections of this mode of thinking in education literature can be seen under similar conceptions. For instance, the pioneers of the Computing Department at Carnegie Mellon University, Alan Perlis and Simon Papert had utilized the same concept as a machinery skill in their teaching. While Perlis (1962) was adopting *'problem-solving methods'* to teach how computers work in his course, Papert (1980) was utilizing *'procedural thinking'* to teach K-12 students LOGO programming language **Figure 1**. As opposed to that, the contemporary approach to CT subtly distinguishes itself from these old conceptions by prioritizing human cognition over machinery skills. Additionally, the development of computational technologies in different forms and their interdisciplinary use have changed the understanding of 'computation = programming' and brought new cognitive skills for the concept of Computational Thinking.

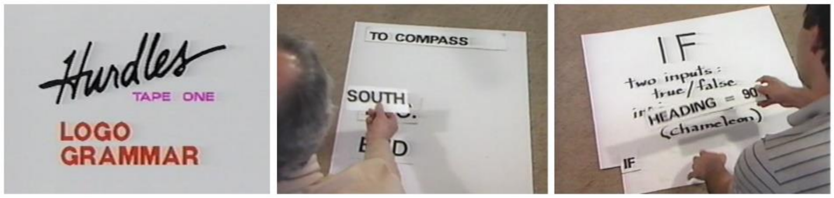

Figure 1: Screenshots from LOGO Grammar Education, Hurdles Video Series (Logo Foundation, n.d.) .

In 2006, the former Head of Microsoft Research Lab and academic, Jeannette M. Wing, known as the first and most influencing person in the education literature (Özçınar, 2017), carried CT to an interdisciplinary level by reintroducing it as a mental tool for human beings. Wing (2017) described CT as "a thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer —human or machine— can effectively carry out" (p: 8). Followingly, Wing (2006, 2008, and 2017) emphasizes the inclusiveness of CT by claiming that this mode of thinking should be for everybody and everywhere, and it should to be included as a fundamental skill in every education curriculum (Wing, 2006). With this new perspective, Wing added more cognitive skills to the definition of CT, and transformed it into a humanly thought. In this regard, CT as a cognitive process involves:

- Making abstractions (the mental tools of computing, necessary to solve the problem),
- Creating layers (problems need to be solved on different levels), and
- Defining relationships between these layers and abstractions (Wing, 2008).

Followingly, Wing's contemporary definition of computational thinking has become the foundation for CT pedagogy around the world (Berthelsen & Nielsen, 2021). It has brought an opportunity to understand and study human-computer interaction on a common ground with a systematic approach. Nevertheless, encouraging this mode of thinking as a combination of skills has also brought challenges in different fields of education. As (Guzdial, 2010) highlight, "spreading computational thinking from computer science to other academic fields – which have their own specialized problem-solving methods– may require adapting existing CT theory and methods to match the needs of "novices" and other non-specialists." In other words, to change how disciplines outside the computer sciences think about and practice computational technologies require to consider how educators think about computational thinking more broadly, so the pedagogical framework can be constructed for such educational reform to succeed. However, since Wing's descriptions of CT (2006, 2008, 2017) did not give much clue about how it can be borrowed by the disciplines outside computer sciences (Hu, 2011) or how these skills can be assessed for different age groups, the new definitions of CT from variety of educational resources came into consideration. In this regard, some of the well-known organizations (CAS-Barefoot Computing, Google for Education, Microsoft Research, EC, NRC, ISTE, TCSA, ACARA) added more cognitive skills to the CT terminology, so the educators could contextualize them in a disciplinary framework **Table 1**.

Table 2: Comparison of Concepts that are used in CT Terminology 2010-2018.

Concepts of CT	Google for Education 2018	National Research Council 2010	CSTA	ISTE 2011	ACARA 2013	Selby and Woollard 2013	Wing 2010	Gülbahar and Kalelioğlu 2016	European Union Report	Microsoft Research Council 2018
Data Collection	X		X	X						
Data Analysis	X									
Generalization	X		X	X		X				
Pattern Recognition	X	X	X	X	X	X	X	X	X	X
Decomposition	X	X	X	X	X	X	X	X	X	X
Abstraction	X	X	X	X	X	X	X	X	X	X
Problem-Solving	X	X	X	X		Affiliated with AT	X	X	Affiliated with AT	
Algorithmic Thinking	X	X	X	X	X	X	X	X	X	X
Data Modeling	X									
Simulation	X									
Parallelization	X									
Debugging										
Data Representation	X									
Automation	X									

Over the last decade, educators from different fields plan their course learning objectives and outcomes to endorse the skills that are affiliated with computational thinking. And in order to match the 21st century learners' attributes at all levels, many other private and public organizations such as National Science Foundation, British Royal Society, European Commission, Google for Education, and Microsoft Research Lab, present online/offline CT education materials that targets the core cognitive skills of CT (Abstraction, Decomposition, Pattern Recognition, and Algorithmic Thinking).

On the other hand, the encouragement of CT use as a mental tool in higher education requires more than a mere endorsement of its cognitive skills with in-class activities. As Berthelsen & Nielsen (2021) states “educational practices cannot and must not focus simply on the acquisition of technical skills and competences, but they must also address the normative question of what the point and purpose of acquiring these skills ought to be”. Educators must encourage their students to ask “What can humans do better than computers, what can computers do better than humans,” and “What is computable” (Wing, 2006, p: 33) and address the answers towards the learning outcomes of their courses. Hence, for learning groups with different cognitive abilities in higher education, the cognitive aspects of CT are usually contextualized under the pedagogical outcomes of the affiliated programme; so, educators and students can find their own creative methods to use CT as a mental tool towards their disciplinary framework. And suffice to say that, each generation require updates in the educational practices and the pedagogical framework of CT education due to unique characteristics in their learning preferences and needs.

3. CONTEXTUALIZING COMPUTATIONAL THINKING FOR GEN-Z IN ARCHITECTURE EDUCATION

Starting from the early stages, the 21st century architecture students engage with digital design tools and platforms throughout their education. They are expected to work creatively, effectively, and efficiently with digital technologies by avoiding computationally expensive solutions in their designs. And, this kind of engagement requires a new set of workflows and behaviors for students and educators (Doyle & Senske, 2017). In other words, the active use of design technologies in different forms require both students and design educators to be competent at computational thinking.

Although ‘the digital technology competence’ is considered as a fundamental skill in the Student Performance Criteria by the Architectural Accreditation Boards (NAAB, 2009), there is not much of information how students’ computational thinking skills would be supported or assessed towards their competence in the digital technology use. Nonetheless, when it comes to CT’s pedagogical alignment in architecture education, different strategies come into consideration. Despite its cognitive contribution to student's thinking process, this mode of thinking has not been acknowledged as a critical thinking skill in the architecture education pedagogy (Doyle & Senske, 2017); and, the conception of computational thinking is not necessarily introduced as mental tool for architecture students.

In most of the architecture schools, students' digital technology competence is affiliated with their computer literacy or technical skills in Computer-aided design technologies; and usually these skills are covered under Computer Aided Design courses. For instance, while Architecture Department at UNC Charlotte endorse programming and computing courses in their third-year curriculum (Senske, 2014), MIT School of Architecture introduces computation within a design studio experience in their first-year curriculum (MIT Architecture, n.d.). Also, several other schools encourage students' computational skills towards generalized programme coding courses. "With the exception of a few schools where "digital" or "paperless" studio experiments were undertaken, the advent of digital analysis and media options has been addressed with specialized course content focused on acquiring skills with the new tools, each one an option on top of the traditional structure" (Johnson, 2016; p:186-187). However, encountering computer literacy in the current pedagogical agenda does not necessarily endorse students' use of CT as a mental tool in their design studies. And this much of variety in the implementation of computational design practices shows that computational thinking education and design education still seems separated by pedagogical gaps and teaching mindsets. Needless to say, leaning on students' competence in digital technologies for computational thinking education will be more problematic for design educators in the near future. In the following years, those born between the mid-1990s and late-2000s, also known as Generation Z students will form the main bulk of students in architecture education.

Millennials (born 1980-1995) and Generation-Z (born 1995-2005), students who born since 1980's are called "digital natives" in literature (Prensky, 2001). Both experienced digital technologies when growing up; but the way that they experienced technology makes significant differences in their learning preferences and attributes. And as opposed to Millennials, Generation-Z's engagement to technology as part of their everyday life makes them 'true digital natives.' Since

"Students from each generation possess specific and unique characteristics due to the circumstances they grew up in and these characteristics affect their perception of formal learning" (Chicca & Shellenbarger, 2018), the distinct characteristics of Generation have necessitated rethinking CT educational practices, pedagogies and teaching approach to meets their learning needs and preferences. And yet, current architecture education pedagogy is already failing to support millennial students' use of computational thinking as a mental tool. According to Doyle and Senske (2017), the presumption that the millennial students who have grown up with digital technologies possess special aptitudes or insights which may be disruptive to learning computing, caused anxieties and biases in the use of digital design technologies, and led to gaps in architectural pedagogy; eventually as digital tools were misunderstood and misappropriated by students and teachers alike (p:193). Hence, within the current framework, Gen-Z students are likely to fail to develop an understanding of cognitive workflows in digital design technologies and intervene them in creative ways because the current architecture education pedagogy is not aligned with the goal of computational thinking learning; instead, it is aligned with the student's computer literacy.

4. DISCUSSION

Gen-z students are skill focused, but also, they embrace lifelong learning skills, which can be updated and advanced with repetition. Hence, instead of training design students towards the hard skills of computing, educators need to focus on the soft skills of computational thinking. Contextualizing the cognitive aspects of CT by targeting similarities between the cognitive aspects of computational thinking and design thinking, such as abstraction, pattern recognition, and decomposition in design classes would help students to internalize and practice these cognitive skills towards hands-on activities. This teaching approach can be seen widely in architectural design studios within the context of different computational design models, such as shape grammar and parametric modeling exercises. Additionally in some cases, design educators claim that this approach contributes students visual and spatial thinking skills as well. However, the cognitive contributions of this approach seem problematic for the Generation Z students CT education. Because Gen-z students prefer autonomous learning, they are self-directed, and independent with a freedom of what/how they learn. They prefer individualized learning (e.g., flexible schedules), engaging and visual learning environments (Chicca & Shellenbarger, 2018). “They prefer individual learning to focus, set their own pace, and make meaning of their learning before having to share that meaning with others” (Seemiller & Grace, 2017). But also, they need a self-assessment and an instant and individual feedback for their actions while learning (Seemiller & Grace, 2017). Hence, the disadvantages of these hands-on computational activities for Gen Z’s CT learning can be seen at this point. The current applications of this approach in literature shows that the hands-on computational activities are unlikely to provide measurable outputs for educators to conduct individual learning experiences for their students with different cognitive abilities. Also, the students cannot receive feedback on their actions for a self-assessment.

Alternatively, another teaching approach offers practicing computational thinking with visual programming tools. In this case, to encourage CT as a mental tool for learning groups with different cognitive abilities, the learning experience of CT skills must be offered in a contextual framework. For that, design educators prefer working with the platforms for visual computing. On the other hand, in case of the insufficient amount of technological equipment and shortage in the teaching staff with a prior experience, educators prefer standardized/generalized course exercises in crowded classrooms and oversee the individual learning needs of their students. Thus, even though students can receive instant feedback for their skills, educators may face challenging to evaluate students’ cognitive skills their individual needs.

5. CONCLUSION

Although the active use of computational technologies in compulsory education highly depends on the cultural, social and economic differences among regions (Czerkawski & Lyman, 2015), CT is shaping the near future of technology education and demanding from different disciplines endorse its cognitive skills. Hence, architecture education’s approach to computational thinking deserves a second thought for the education of Gen-Z. Regarding that, this article summarized the current pedagogical agenda of computational thinking in architecture education, and addressed some of the challenges to be solved for Gen-Z's CT education. It discussed that to advance Gen-Z students' understanding of cognitive workflows in digital design technologies and endorse them to use these technologies in creative ways because the current architecture education pedagogy must be aligned with the goal of computational thinking learning, not with the student's computer literacy. It also showed that to diagnose students individual learning needs toward differences in their cognitive abilities and education level, educators must be aware of choosing appropriate assessment methods and tools for teaching computational thinking.

For the next years, the theoretical frameworks for Generation Z's learning attributes must be considered in designing future pedagogies for the alignment of computational thinking to architectural education. Additionally, in order to include the cognitive contributions of computational thinking in the programme outcomes of architecture education, more studies are required for educators encountering CT to their pedagogical agenda.

References

Berthelsen, U. D., & Nielsen, C. F. (2021). Democracy and computation: a normative perspective on the magic of the new millennium. *Computational Thinking in Education*, 57–72, Routledge.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Exploring the field of computational thinking as a 21st century skill. *EDULEARN proceedings*. https://doi.org/10.21125/edulearn.2016.2136

Cecutti, L., Chemero, A., & Lee, S. W. (2021). Technology may change cognition without necessarily harming it. *Nature Human Behaviour*, 5(8), 973-975. https://doi.org/10.1038/s41562-021-01162-0

Chicca, J., & Shellenbarger, T. (2018). Generation z: approaches and teaching-learning practices for nursing professional development practitioners. *Journal for Nurses in Professional Development*, 34(5). https://doi.org/10.1097/NND.0000000000000478

Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57–65. https://doi.org/10.1007/s11528-015-0840-3

Doyle, S., & Senske, N. (2017). Between design and digital: bridging the gaps in architectural education. *Charrette*, 4(1), 101–116.

Dror, I. E. (2011). Brain friendly technology: What is it? And why do we need it? *Technology Enhanced Learning and Cognition*, 27, 1.

Google for Education. (2019). *Resources*. Retrieved January 7, 2019, from https://edu.google.com/resources/programs/exploring-computational-thinking/#!resources

Guzdial, M. (2010). Does contextualized computing education help? *ACM Inroads*, 1(4), 4-6. https://doi.org/10.1145/1869746.1869747

Hu, C. (2011). Computational thinking: what it might mean and what we might do about it. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 223–227. https://doi.org/10.1145/1999747.1999811

Johnson, B. R. (2016). Design computing: An overview of an emergent Field. *Routledge*. https://doi.org/10.4324/9781315680057

Logo Foundation. (n.d.). *Logo resources*. https://el.media.mit.edu/logo-foundation/resources/index.html. Retrieved August 23, 2019, from https://el.media.mit.edu/logo-foundation/resources/index.html

Microsoft Education. (2019). *Definition of Computational Thinking*. https://education.microsoft.com/Story/Course?token=4dFPG, last access 07.01.2019.

MIT Architecture. (n.d.). *Undergraduate programs*. Retrieved May 20, 2020, from https://architecture.mit.edu/undergraduate-programs#first-year-subjects

NAAB. (2009). *Substantial Equivalency*. Retrieved February 11, 2019, from ttps://www.naab.org/international/substantial-equivalency/

Özçınar, H. (2017). Hesaplamalı düşünme araştırmalarının bibliyometrik analizi. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(2), 149–171.

Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology*, 3(3), 249-262. https://doi.org/10.1080/0020739700030306

Perlis, A. (1962). The Computer in the University. Computers and the World of the Future, 180-219.

Prensky, M. (2001). Digital natives, digital immigrants Part 1. *On the Horizon*, 9(5), 1-6. https://doi.org/10.1108/10748120110424816

Seemiller, C., & Grace, M. (2017). Generation Z: Educating and engaging the next generation of students. About Campus: Enriching the Student Learning Experience, 22(3), 21-26. https://doi.org/10.1002/abc.21293

Senske, N. (2014). Digital minds, materials, and ethics: Linking computational thinking and digital craft. *CAADRIA proceedings*. https://doi.org/10.52842/conf.caadria.2014.831

Shorey, S., Chan, V., Rajendran, P., & Ang, E. (2021). Learning styles, preferences and needs of Generation Z healthcare students: Scoping review. *Nurse Education in Practice*, 57, 103247. https://doi.org/10.1016/j.nepr.2021.103247

Simon, H. A., & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2), 145–159. https://doi.org/10.1037/h0030806

Weisberg, R. W., & Reeves, L. M. (2013). *Cognition: from memory to creativity*. John Wiley & Sons.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.

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

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14. https://doi.org/10.17471/2499-4324/922