Adv Mobile Learn Educ Res, 2024, 4(2): 1224-1241
DOI: 10.25082/AMLER.2024.02.015

RESEARCH ARTICLE

Study on the implementation of the CAL (Coding as Another Language) curriculum in early childhood education

Georgia E. Kalyva

Department of Preschool Education, Faculty of Education, University of Crete, Crete, Greece

Correspondence to: Georgia E. Kalyva, Department of Preschool Education, Faculty of Education, University of Crete, Crete, Greece;
Email: kalivageo@gmail.com

Received: September 25, 2024;
Accepted: December 12, 2024;
Published: December 18, 2024.

Citation: Kalyva, G. E. (2024). Study on the implementation of the CAL (Coding as Another Language) curriculum in early childhood education. *Advances in Mobile Learning Educational Research*, 4(2), 1224-1241. https://doi.org/10.25082/AMLER.2024.02.015

Abstract: In recent years, Information and Communication Technologies (ICT) have been increasingly integrated into preschool education, enriching the learning process with valuable information and motivation while capturing young students' interest. Numerous studies emphasize the importance and impact of utilizing ICT in education, introducing new dynamics to learning experiences, particularly as children grow up surrounded by diverse digital stimuli. From an early age, they interact with applications and digital tools, especially touchscreens, building digital experiences that become essential in educational interventions. Rapid technological advancements have led to the development of a wide range of applications that enhance educational processes, significantly influencing how young children learn and process information. Games, mobile devices, interactive whiteboards, and programming applications aid in the development of critical ICT skills, logical thinking, associative reasoning, computational thinking, and programming abilities. Striking the right balance between play and learning creates a rich, stimulating environment that fosters young learners' growth in skills and abilities. Considering the European Union Council's recommendation of May 22, 2018, digital competence is one of the key skills that must be cultivated both within and beyond school throughout life. As future citizens, today's students need skills that will enable them to solve problems effectively. Our digital society demands individuals capable of responding efficiently to challenges across various social contexts. Consequently, teaching students computational thinking and programming skills has become essential. Research demonstrates that preschool children, when faced with problem-solving tasks requiring the creation of code, can develop basic programming and computational thinking skills, such as debugging and understanding sequences. Moreover, in recent years, digital applications specifically designed to teach young children fundamental programming concepts have emerged. While numerous learning environments focus on coding skills, studies on their effectiveness in fostering coding and computational thinking in preschoolers remain limited. As the number of mobile learning applications grows, it is crucial to identify those with genuine educational value, avoiding those marketed as educational without substantive merit. This study describes an educational intervention based on experiential and collaborative teaching principles. The research aims to demonstrate that through the use of the CAL (Coding as Another Language) approach within the ScratchJr programming environment, significant programming and computational thinking skills can be cultivated in a conventional Greek preschool classroom. The teacher and researcher are the same individual, ensuring direct involvement in the educational process.

Keywords: pre-primary education, kindergarten, computational thinking

1 Theoretical framework

1.1 ICT in preschool education

In today's world, digital literacy has become a fundamental aspect of daily life. Children today are born into a digitally enriched environment. Even before they develop basic reading and writing skills, they interact with multimodal environments where the "screen" is an essential tool in everyday life. Through observation and imitation, digital devices become easy-to-use tools for young children, as many parents welcome the opportunities technology offers their children (Chaudron, Gioia & Gemo, 2018). Watching children be captivated by interacting with a touch screen naturally leads to the thought that there is a need for a more systematic and organized exposure to the vast amount of information available. Proper guidance is necessary to help them navigate towards meaningful information and knowledge. Research has shown that using technology in teaching, particularly in subjects like mathematics and geometry, leads to the development of advanced mathematical thinking and reasoning skills. In a study conducted in Greece during the 2013-2014 school year with 241 kindergarten children aged 4 to 6, the results were impressive (Papadakis et al., 2016). Based on the Realistic Mathematics Education (RME) approach, this research involved teaching interventions using activities that connected mathematical problems to the real world, providing motivation for children to interact with their surroundings and solve problems.

The findings of the study confirmed the researchers' hypotheses, demonstrating that connecting mathematics to real-life situations and activities that merged the physical and digital world improved children's mathematical thinking. The experimental group, which participated in teaching interventions, performed significantly better than the control group in all areas of mathematical thinking. It is evident, therefore, that cultivating computational thinking systematically through targeted tools is essential. An interesting study conducted in Florida by Reeves et al. (2017) involved 28 kindergarten students aged 4 and 5 in activities aimed at enhancing early literacy and mathematical skills. The findings showed that mobile learning, combined with informal feedback, significantly boosted the phonological awareness and math skills of children using iPads, compared to a control group that did not receive technology-supported instruction. Similarly, the research of Zarani and Alexandraki (2018) confirmed that teaching mathematics (specifically multiplication) using mobile devices was significantly more effective in helping children understand the subject compared to traditional teaching methods. Additionally, Hatzigianni et al. (2018) concluded that mobile devices can even enrich free play in kindergartens, without replacing it. Children used tablets to record their play activities (e.g., taking photos, recording videos), showing great enthusiasm. A recent study by Papadakis et al. (2021) confirmed earlier research findings, indicating that teaching mathematical concepts using mobile devices in kindergartens leads to significantly improved learning outcomes. Another perspective was offered by Schriever (2021), who explored the role of kindergarten teachers in mediating the use of ICT and mobile devices. Teachers often found themselves managing expectations or concerns from parents regarding the use of technology in the classroom.

Italian educators participating in a study also recognized the advantages of mobile devices in the learning process, such as their portability and ability to facilitate activities outside the classroom. However, they also expressed concerns about potential technical issues, such as internet access difficulties or coordinating the use of multiple devices in the classroom (Dovigo, 2021). In line with the kindergarten curricula, young students are engaged in various activities aimed at understanding basic mathematical concepts. As the area of computing has been incorporated into the curriculum, ICT has become a valuable tool for enhancing and enriching the educational and learning process in all its forms. Mobile learning (m-learning), in particular, is now a key component in preschool education (Papadakis, Alexandraki & Zaranis, 2022). The observation of children's fascination with touch screen devices reinforces the idea that such interaction can significantly contribute to their learning process, especially in mathematical thinking. In recent years, with the increasing integration of digital tools in schools, the positive correlation between ICT and the enhancement of children's mathematical thinking and abilities has become clear (Panagiotakopoulos et al., 2013). According to McManis and Gunnewig (2012), ICT-enhanced teaching, when the educator maintains a supportive role, greatly enhances the skills that students acquire in almost all areas of mathematics. Papadakis et al. (2016) highlight the importance of ICT in mathematics teaching in kindergartens, noting that mobile devices, due to their interactive nature, seem to improve learning outcomes compared to traditional digital activities using computers (Vaiopoulou et al., 2021).

1.2 Conceptual approach to computational thinking

The term "Computational Thinking" (CT) refers to a fundamental skill that extends beyond computer science and is applied to many fields of human life and everyday activities. It encompasses a set of cognitive tools that are essential in areas ranging from mathematics and physics to the social sciences and the arts (Kalogiannakis & Papadakis, 2020). In the 1980s, the renowned mathematician and computer scientist Seymour Papert first introduced the term "Computational Thinking" to establish a framework for understanding children's interactions with computers (Papert, 1980). In March 2006, Jeannette Wing revisited the concept in her article "Computational Thinking," emphasizing its importance as a core human skill relevant to many aspects of life (Wing, 2006). In 2017, Valerie J. Shute, Chen Suna, and Jodi Asbell-Clarke, in a detailed study, expanded the definition of computational thinking, highlighting its importance as a conceptual foundation for solving problems algorithmically, with or without the use of a computer, and creating solutions that can be reused in different contexts (Shute et al., 2017). That same year, Román-González et al. (2017), after reviewing the literature, identified three categories of CT definitions: general definitions, functional definitions, and educational definitions. These categories emphasize different aspects of computational thinking, making it a versatile tool for developing cognitive skills and solving problems across various fields.

In the general definitions, we can categorize the one by Wing (2006), which states that human cognition and computers collaborate to solve a problem situation. In 2011, Wing formulated a second definition, also classified under general definitions. According to this, computational thinking is governed by cognitive processes such as problem formulation and solution formulation in a way that allows processing by a specific information processing medium (Wing, 2011).

Regarding the second category, operational definitions, it is worth mentioning the research conducted under the initiative of the International Society for Technology in Education (I.S.T.E.) and the Computer Science Teachers Association (C.S.T.A.), according to which a set of skills, attitudes, and behaviors aimed at the immediate and effective resolution of a problem situation when interacting with computer science describes the precise meaning of computational thinking. Indeed, problem formulation, logical organization and analysis of information and data, their abstract representation, the activation of algorithmic thinking leading to solutions, the examination of the existence or non-existence of potential solutions, and the testing of their application, as well as the ability to generalize the entire process to other contexts, are the key characteristics of computational thinking based on its operational definition (Papadakis et al., 2021). As for the educational definition of computational thinking, as the term itself suggests, it refers to the cultivation of this skill within the school context. A constructive approach to understanding and developing students' skills, based on this definition, is the framework proposed by Brennan and Resnick (Brennan & Resnick, 2012). This framework is based on three dimensions:

(1) Computational Concepts: Basic concepts used in programming language (sequence, repetition, conditions, variables, function).

(2) Computational Practices: The practices and methods used during programming (design and testing with iterative capabilities, debugging, creating and recognizing patterns, collaborative work towards a common outcome).

(3) Computational Perspectives: The sense of completeness that students feel regarding the world of Technology (empowerment, expression, connection...).

Finally, in 2018, Bers presented computational thinking as a process that offers opportunities for expression and creativity, and this very expression and creativity possess such "power" that it leads effortlessly and, primarily, enjoyably to the resolution of problem situations. According to Bers, there are seven powerful ideas of computational thinking for preschool age:

(1) Algorithms;
(2) Modularity;
(3) Representation;
(4) Control Structures;
(5) Hardware/Software;
(6) Debugging;
(7) Design Process.

It is impressive to note that in 1980, the term "powerful ideas" was first used by the pioneer Seymour Papert (Papert, 1980), who identified that when someone engages in activities that are particularly interesting and significant to them and serve as a source of inspiration, they are led to new ways of thinking and problem-solving characterized by innovation and creativity.

1.3 Programming in preschool education

The always pioneering Seymour Papert, very early on, in 1980, referred to the valuable contribution of programming skills when acquired through a specific educational process governed by design and planning. Trials and experiments in creating new objects through engagement in activities that have meaning for the child are, according to Seymour Papert, the guarantees that lead them, through discovery learning, to the attainment of substantial knowledge and, thus, effective learning. In an effort to enhance the mathematical abilities of young students and introduce them to programming through enjoyable and creative activities, he created the first programming language for children, known to those involved in Technology in Education (TIE) as Logo (Papert, 1980). The Logo programming environment encourages children to program the "turtle," a virtual robot that can draw as it moves across the screen. Most research conducted to investigate programming skills in preschool age has utilized the Logo programming environment. Nowadays, observing small children fascinated by interacting with a touchscreen, it is natural to think that this method can serve as a boost in the learning process of kindergarten across all learning subjects while also activating the associative thinking and capabilities of young children. In recent years, as TIE has been supported by digital equipment in schools, a positive correlation between TIE and the enhancement of programming skills, mathematical thinking, and abilities of children is evident (Panagiotakopoulos et al., 2013). According to McManis and Gunnewig (2012), teaching enhanced by TIE, with the educator maintaining a supportive role, significantly increases the skills that students acquire.

However, Fessakis et al. (2013) consider the design and implementation of targeted teaching interventions with engaging teaching practices as essential tools for integrating programming into kindergarten rather than merely relying on various digital learning programming environments. Empirical research (Macrides, Miliou & Angeli, 2021) has concluded that fundamental concepts of computational thinking can indeed be cultivated and enhanced in preschool students through the integration of different learning programming environments in the educational process.

1.4 Curriculum and computational thinking

It is believed that since the time of the pandemic, the educational community has invested more in developing programming skills and computational thinking, including through the curricula. During that extraordinary period, the immediate need for remote education arose (Papadakis & Kalogiannakis, 2019; 2020). The conditions that emerged brought to light a new term, "Emergency Remote Teaching," which refers to education conducted from a distance under emergency conditions. Teachers and students in the country found themselves in a "digital upheaval" with political and economic implications, particularly regarding the provision of equipment to facilitate the uninterrupted process of distance education and the implementation of curricula.

During the remote education process, educators had already engaged in actions to develop their digital skills. Several researchers at that time focused on investigating the digital skills of teachers in utilizing Information and Communication Technologies (ICT), as these are an integral part of distance education (König et al., 2020). Teachers needed to refer back to the curricula and find ways to connect them with the new digital data they were called upon to address! In the revised curriculum (2014) for preschool education, ICT represents a distinct learning area that operates complementarily to others, aiming primarily to introduce young children to digital literacy and to establish a foundation for creating knowledge, attitudes, values, and perceptions about ICT (IEP, 2014). The development and cultivation of essential digital skills, programming, and logical and associative thinking in preschool students, through the integration of computational thinking in the curricula, represent a relatively modern approach. With the primary aim of the curricula for Greek kindergartens being the enhancement of digital literacy among young students, the use of ICT is supported through attractive and well-designed educational situations (Ministry of Education, 2021).

Pattern recognition, decomposition, algorithmic thinking, and abstraction are skills that enhance algorithmic thinking. Through interactive games and applications, Robotics, programming games, role-playing, design creation, and patterns, as well as through disconnected activities or those enriched with ICT, these skills can be cultivated. Particular importance in the curricula for kindergarten is the involvement of goals across scientific and thematic fields. This allows for the integration of digital tools into all Thematic Fields, with the main aim being information and digital literacy on one hand, and multiple benefits for developing computational thinking on the other. Indeed, in the most recent curriculum for preschool education (Ministry of Education, 2021), the thematic unit of ICT includes three subunits that interact and complement each other, providing significant flexibility in their application in the school setting. These subunits are presented as: a) Introduction and Communication with ICT b) Discovery, Programming, and Digital Play c) Information Processing and Digital Creation It is evident that the curriculum for kindergarten supports a developmentally appropriate digital environment, recognizing ICT as a powerful learning and developmental tool for computational thinking, allowing for the use of tools that assist in concept processing and addressing communication difficulties, activating expression and creativity potential, and responding to specific characteristics and needs. As a result of all the above, the development of innovation in children's creative thinking occurs, enabling them, through enjoyable communicative situations with educational content, to lead to problem-solving by activating their computational thinking. What is required is the targeted design of educational scenarios to ensure the activation of motivation and interest among young students from the outset.

1.5 Learning programming environments

1.5.1 Logo programming environments

Logo environments are based on the first programming language created to enhance the mathematical abilities of young learners and help them understand programming processes through engaging and creative activities. This is the programming language for children, Logo (Papert, 1980). The Logo programming environment encourages children to program a "turtle," a virtual robot that can draw as it moves across the screen. Even today, it remains a source of inspiration for the creation of new programming systems (Fessakis et al., 2019).

1.5.2 Floor Robots

Bee Bot: The Bee Bot is a very popular robot found in schools, shaped like a bee. Its attractive appearance and relatively simple operation make it instantly loved by young children. It moves on a floor surface divided into squares measuring 15x15 cm and can accept over 40 commands. On its top, it features navigation buttons (forward, backward, left turn 90°, and right turn 90°). Additional buttons include X (clear), which clears the memory of previous commands; GO, which starts executing the commands; and II (pause), which pauses the execution of commands. Upon completion of its commands, the little bee alerts users with a sound and lights up its eyes (Figure 1).

Figure 1 Bee Bot

Blue Bot: This is another floor robot that can store up to 200 commands! It closely resembles the Bee Bot but has a transparent shell that allows children to visually interact with its interior. The command buttons are located on the top of its body, and its usage is similar to that of the Bee Bot. The added value of the Blue Bot, beyond the visual interaction that captivates children, is the capability for wireless connectivity with mobile devices and computers (Figure 2).

Figure 2 Blue Bot

Colby Robot Mouse: Visually impressive and accessible to young children, this small floor robot shares common operational features with the Bee Bot and Blue Bot. It is programmed using colorful arrow-shaped navigation buttons located on its top. Each direction has a different color. It also includes a button for deleting stored commands and a button that initiates the execution of the provided commands. The robot is accompanied by square pieces that can be connected to create the path along which it moves. Notably, children can easily create scenarios for the "life" of the robot, shaping the space in which it moves with additional pieces to create mazes, tunnels, castles, etc. The robot can incorporate sound and offers two movement speeds, allowing for use on both grid tracks and other surfaces (Figure 3).

Figure 3 Colby Robot Mouse

Kids Bits Coding Robot: An attractive programmable robot that moves on a flat surface, like the previous robots, it does not require wiring and operates on a rechargeable battery. It features command buttons to control its sensors and motors, enabling it to draw on the surface during movement and connect to mobile devices via Bluetooth with a specific application. Young children are excited to create their scenarios, programming it, for example, to avoid obstacles (Figure 4).

Figure 4 Kids Bits Coding Robot

Code-a-Pillar Robot: This caterpillar-shaped robot is visually appealing and attracts the interest of young children. It features sound and movement. It consists of a fixed piece (the head), while the other detachable pieces form its body, each having a pre-programmed movement, as shown in Figure 5. Essentially, the programming of its movement is determined by how the body pieces are arranged, creating the code. As the command is executed, each section of the robot lights up, and movement halts in the presence of an obstacle. The head has a button that starts or restarts the robot's movement (Figure 5).

Figure 5 Code-a-Pillar Robot

1.5.3 Visual programming environments

Visual programming environments refer to those digital learning and computational thinking environments that allow the use of graphical elements and the selection of appropriate digital tiles that perform specific actions, enabling the creation of code through drag-and-drop (Cheng, 2019).

1.5.4 ScratchJr

ScratchJr is a programming environment designed for children aged 5-7. It allows the creation of interactive stories using basic programming concepts. This programming environment emerged from the exceptional collaboration of the DevTech research team at Tufts University and the MIT Lifelong Kindergarten Group. ScratchJr operates in an accessible way: children can choose characters and objects from a library, place them in a scene of their choice from the program's library, and then create logical connections between the objects and characters. It offers six categories of commands: yellow start tiles, blue movement tiles, purple appearance tiles, green sound tiles, orange control flow tiles, and red end tiles (Figure 6). Interaction between objects and characters is achieved using programming tiles, which are selected and moved to the programming area where they are activated. The connection between the tiles creates program codes that are executed from left to right (Bers, 2018).

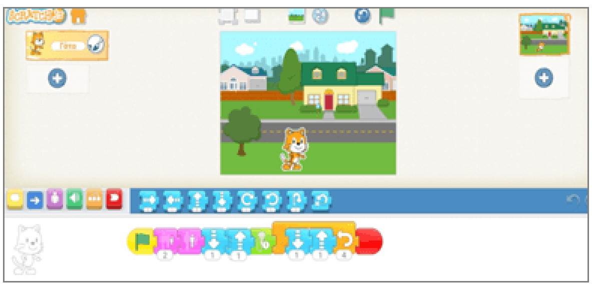

Figure 6 ScratchJr programming environment

In this way, children discover the fundamental concepts of programming and express their creativity within a playful digital environment, which can later be extended to physical activities or vice versa.

Daisy the Dinosaur: This is a free educational programming application compatible with the iOS operating system. Designed to help young children develop basic programming skills in a fun way, it offers missions with varying difficulty levels that require children to use drag-and-drop actions. Through this, they grasp the concepts of sequence, loops, and conditionals. By performing actions with a finger touch on the screen, children program the character Daisy, a cute little dinosaur, to follow the instructions they have provided (Figure 7 and 8). Upon completion of an action, they receive a reward message!

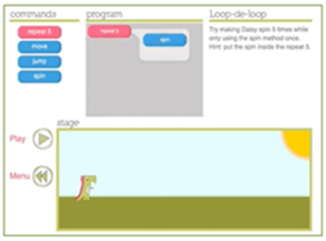

Figure 7 Daisy programming environment

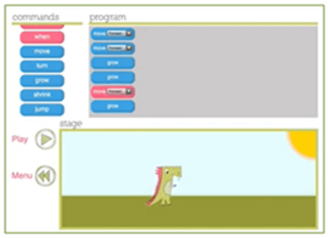

Figure 8 Dinosaur programming environment

Code.org: Code.org is a non-profit platform that promotes education and the development of programming skills. This platform offers free lesson programs, games, and programming resources covering various educational levels and skill sets. Notably, it has turned modern digital games and scenarios, particularly appealing to children, such as Minecraft, into digital programming activities, immediately capturing children's interest. Its operation focuses on creative learning through interactive games and activities. The entire environment is designed to be user-friendly, providing a learning experience that is both enjoyable and educational. Through Code.org, both children and adults can familiarize themselves with the fundamental principles of programming and develop their skills in this field in a fun way while enhancing computational thinking. The tiles are again the technical features used to create the necessary code for the child's response to the activity, with the interactive environment being visually enriched with colors and other structural elements.

1.6 Educational robotics: A new learning tool

The playful approach to learning subjects within an environment of creativity and autonomy forms one of the foundations of effective education. This philosophy was adopted by Bers and her team in 2014, proposing the introduction of activities that enhance the development of computational thinking skills in the school context. In recent years, practical applications and research related to educational robotics have highlighted its importance as a tool that supports learning and enhances students' programming skills. Below is a presentation of some of the most popular educational robots and kits that contribute to STEM education.

1.6.1 KIBO robotics kit

KIBO is the advanced version of the educational toolset KIWI. It includes connectable wooden blocks with barcodes, sound and light sensors, wheels, and motors. Children use the wooden blocks to create sequences of commands that determine the robot's behavior (Figure 9).

Figure 9 KIBO Robotics Kit

1.6.2 LEGO education WeDo

LEGO Education WeDo is designed for children aged 7-11, providing them with fundamental concepts of technology, engineering, and mathematics through the construction and programming of robotic models. The kit includes LEGO building elements, sensors, motors, and a computer or tablet with programming software. The process involves assembling models, connecting sensors, programming, and executing programs, encouraging creativity and critical thinking (Figure 10).

Figure 10 LEGO Education WeDo

1.6.3 LEGO education SPIKE

Designed for students aged 10 and above, LEGO Education SPIKE includes LEGO building elements, sensors, and a powerful programmable hub. The SPIKE App software offers a graphical programming environment, supporting both Scratch and Python, enhancing collaboration and creativity among students (Figure 11).

Figure 11 LEGO Education SPIKE

1.6.4 Ozobot

Ozobot is a small robot that introduces children to the fundamental principles of programming and robotics. It can be programmed using platforms such as OzoBlockly and can interact with its environment, making it ideal for educational settings (Figure 12).

Figure 12 Ozobot

1.6.5 Dash & Dot

The Dash & Dot family offers educational robots that teach programming through interactive applications. Dash is more advanced, while Dot interacts with its surroundings. These robots enhance logical thinking, problem-solving, and collaboration (Figure 13).

Figure 13 Dash & Dot

1.6.6 mBot and Arduino robot car

The mBot, based on the Arduino platform, provides a hands-on introduction to robotics. It features sensors and motors that allow it to interact with its environment. It is a powerful tool for education in the principles of robotics and programming (Figure 14).

Figure 14 mBot and Arduino Robot Car

1.6.7 Turtle robot

The Turtle Robot comes from the Logo programming language and allows children to learn basic programming principles by executing commands given with colored cards. It is user-friendly and promotes collaboration among students (Figure 15).

Figure 15 Turtle Robot

1.6.8 Cubetto

Cubetto is a wooden robot that executes commands via a programming board. Children can create programs in simple ways, enhancing learning through play (Figure 16).

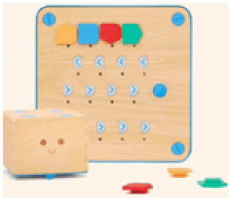

Figure 16 Cubetto

1.6.9 Sphero mini

Sphero Mini is a spherical robot that offers numerous programming and interaction possibilities with the environment. Sphero apps facilitate learning and programming (Figure 17).

Figure 17 Sphero Mini

1.6.10 Social robot (SoRo) toolkit

The SoRo toolkit allows children to interact socially with a robot, experimenting with concepts of computational thinking. Through this process, both interpersonal and computational skills are developed (Figure 18).

Figure 18 Social Robot (SoRo) Toolkit

1.7 The use of the CAL approach for fostering computational thinking in early childhood education

The "Coding as Another Language (CAL)" approach refers to a new curriculum inspired by the pedagogical views and philosophy of Professor Marina Bers, as described in her books "Coding as a Playground" and "Beyond Coding: Teaching Human Values to Children." According to this framework, four key elements combine to create a comprehensive and enriched approach to teaching coding and technology that not only develops children's technical skills but also enhances their social and emotional capacities.

1.7.1 Creativity

Young children need space for free expression of their ideas and the development of their imagination. They require both space and motivation to connect their ideas and imagination expressively through their creativity. The motivation for expressing creativity comes from carefully and thoughtfully designed activities that engage students in projects where they can create and experiment with technology in a safe and developmentally appropriate environment, allowing them to try out new ideas and solutions.

1.7.2 Coding as a playground

In every educational process, we need to create "playgrounds" in which children can discover coding through play, joy, and satisfaction. Coding is not merely a technical skill; it is also a creative and enjoyable activity where play takes a leading role, making learning more engaging and enjoyable.

1.7.3 Human values

It is possible to develop digital and technical skills while simultaneously enhancing humanistic values. Not only is this feasible, but it is also admirable! Empathy, collaboration, responsibility, and other important human values are significantly strengthened when children, from a young age, learn to use technology in ways that benefit society and those around them.

1.7.4 Personal expression

Children's personal creations, especially when presented as unique and wonderful, provide enriched stimuli and support that enhance their desire for further expression of their interests and significant uniqueness through their projects. Personal expression, which recognizes autonomy as an ally and companion in the development of children's skills and abilities, is considered an essential element of learning, allowing children to connect new information with their personal experiences and develop a sense of ownership over their work. The "Coding as Another Language (CAL)" approach, developed by the DevTech Research Group, supports the application of strong concepts in computer science that, according to Bers (2018), include: algorithms, modularity, representation, control structures, hardware/software, debugging, and the design process.

The literature review indicates that, in developmentally appropriate environments, whether physical or digital, the educational process yields excellent learning outcomes. This is further reinforced when developmentally appropriate environments are used to foster programming, computational thinking, and coding skills (Macrides et al., 2021). The implementation of the CAL program is based on the use of ScratchJr software. This is a well-known programming environment for introducing and reinforcing basic concepts of computer science and coding to preschool students. According to the literature review, the use of ScratchJr as a means for fostering programming and computational thinking skills goes beyond enhancing the aforementioned strong concepts (Strawhacker, Lee & Bers, 2018) and emphasizes the development of children's creative abilities (Strawhacker & Bers, 2019) and enhances their associative thinking, leading to successful engagement in problem-solving situations (Papadakis, Kalogiannakis & Zaranis, 2016).

(1) Papadakis et al. (2015), in order to investigate the learning response of young students and the impact of ScratchJr on developing computer science skills, conducted research on 43 students attending a public and a private kindergarten in the region of Crete. After a 7-hour teaching intervention using the software, they concluded that even young kindergarten students, engaged in this type of learning process, developed satisfactory programming concepts, and found that gender did not affect their performance.

(2) In a study involving 62 children from kindergarten to the 2nd grade, the children were encouraged, through ScratchJr, to create projects, stories, and games based on their personal interests, making them meaningful to themselves (Portelance & Bers, 2015). The study also employed student interviews, a developmentally appropriate technique that captures children's interest. The children created projects of great interest, and it was observed that they selected blocks for their projects based on their age group, with movement activation tiles being particularly favored. An essential role was played by the children's ability to act autonomously based on their personal interests.

(3) Strawhacker et al. (2018), in a research study conducted with 200 students attending six schools in the United States, ranging from kindergarten to 2nd grade, aimed at developing skills in identifying and correcting errors in code creation through ScratchJr. They reached the significant conclusion that students' age was correlated with their performance. Older children showed greater improvement in developing programming skills. It is worth noting that in classes where the educational methodology promoted student-centered and inquiry-based learning, the results were notably superior.

(4) Similarly, Strawhacker et al. (2019) found that preschool students, although they ultimately showed improvement in their computational thinking, struggled to understand the

functionality of digital blocks, and it took considerable time for them to create different codes for multiple characters on the same screen.

(5) A relatively recent study conducted by Kyza et al. (2021) involved 51 children aged 6-12, divided into two age groups: 6-9 and 10-12. For 4 days, each group participated in a 6-hour introduction to ScratchJr. Initially, they were asked to program freely within the ScratchJr environment, followed by simple programming exercises and creating digital stories. According to the study's results, older children demonstrated more developed abstract thinking and the ability to make parallels and decompose problems. Furthermore, the code they wrote for creating digital stories was more effective than that of younger children, which is expected since the application primarily targets preschool and primary school children. Younger children, on the other hand, had some non-functional parts in their code and generally did not use control commands, showing a preference for motion commands.

(6) In research conducted by Gaki and Jimogianis (2021) in a public kindergarten involving 18 children, the design and implementation of an intervention using ScratchJr was presented, aimed at solving simple mathematical problems through programming. Evaluation was done through observation and semi-structured interviews, and according to its results, children were able to create simple projects and solve mathematical problems using programming blocks. Furthermore, the difficulties identified by previous researchers (Papadakis et al., 2016), such as managing repeat blocks, timing, and multiple movements, were confirmed. Although the sample size was not large, the researchers successfully presented an effective intervention that showed significant results for integrating programming into early childhood education and its positive effects on understanding mathematical and spatial concepts.

2 Methodology

2.1 Research question

Does the computational thinking of preschool-age students improve with their engagement in a teaching intervention through the application of the CAL program?

2.2 Sample

This research targets preschool-age students, aged 4-6 years, attending a public Kindergarten in the municipality of Agios Nikolaos, in the Lassithi Prefecture. The sample was selected using convenience sampling, as the researcher is a permanent educator at this particular school, facilitating access to the sample. The sample consists of 22 children (both kindergarten and preschool), of which 9 are girls and 13 are boys, all attending the same class in the kindergarten.

2.3 Research procedure

This research was conducted as part of a master's thesis for the Master's Program in "Educational Sciences" at the Department of Preschool Education of the University of Crete and lasted three weeks. The study employed a quantitative research design with pre-test and post-test measures. The CAL program, Coding As Another Language (CAL), designed by the DevTech Research Group, was utilized during the intervention, which took place within the framework of a standard daily program in a kindergarten classroom. During the research, the rules and principles of the ethics code for educational research with children were adhered to (Petousi & Sifaki, 2020). The researcher ensured the anonymity and protection of the participants' personal data. Prior to conducting the research, access to the TechCheck assessment tool (Relkin, Ruiter & Bers, 2020) was obtained. This tool is a valid assessment instrument for evaluating computational thinking and consists of 15 multiple-choice questions administered individually using a portable device, both before and after the intervention. The researcher secured the right to access the tool by applying to DevTech. Following approval, she underwent training and certification for using the tool. The pre-test was conducted in a calm environment during the centers of learning within the classroom, following clear and understandable instructions for the students. Once the students completed the pre-test, the three-week teaching intervention was implemented, and the post-test was subsequently administered using the same assessment tool.

2.4 Theoretical background of the intervention

Piaget (1962) emphasized the importance of play in children's development. ScratchJr provides an environment where children can learn through play, experimenting with new ideas. According to Piaget (1970), learning is an active process through experience. Seymour Papert (1980) argues that learning becomes more effective when learning objects are meaningful to children.

Vygotsky (1978) highlighted the significance of social interaction in learning, proposing that learning is socially constructed. ScratchJr encourages collaborative learning, allowing children to work together. Mayer (2009) states that learning is more effective when information is presented in multiple ways.

In summary, ScratchJr combines the aforementioned theories to create a rich and supportive learning environment, enhancing creativity and understanding of programming principles.

2.5 Measurements

Research Question: Does the computational thinking of preschool-age students improve with their engagement in a teaching intervention through the application of the CAL program?
(1) Independent Variable: The CAL pedagogical approach.
(2) Dependent Variable: Improvement in the computational thinking of preschool-age students.

2.6 Data analysis

In the research study, 22 students aged 4-6 years participated. Of the sample, 40.9% were boys and 59.1% were girls.

Table 1 Student gender

Gender	Frequency	Percent	Valid Percent	Cumulative Percent
Boy	9	40.9	40.9	40.9
Girl	13	59.1	59.1	100.0
Total	22	100.0	100.0	

Table 2 Paired samples test

Pair 1	Paired Differences					t	df	Sig.(2-tailed)
				95% CI				
	Mean	SD	SE	Lower	Upper			
Pretest Score – Posttest Score	-4.000	1.234	0.263	-4.547	-3.453	-15.199	21	0.000

According to the results, the mean score in the pre-test was 8.18 (standard deviation 2.26) and in the post-test it was 12.18 (standard deviation 1.87), indicating a positive impact of the intervention. The results of the analysis show that the difference in mean scores is statistically significant ($t = -15.199$, $df = 21$, $p < 0.001$), confirming the improvement in students' computational thinking.

The teaching intervention through the CAL program had a statistically significant effect on the computational thinking of the students. The mean difference of -4.000 indicates an improvement of 4 points. The very low p-value (0.000) confirms that this difference is statistically significant.

Table 3 Descriptive statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Pretest Score	22	5	12	8.18	2.260
Posttest Score	22	9	15	12.18	1.868
Valid N (listwise)	22				

2.6.1 Evaluating the effectiveness of an educational intervention using the CAL program

The data presented in the table above reveal significant differences in students' performance before and after the educational intervention. Specifically, the average score of children in the pretest was 8.18 (standard deviation: 2.26), whereas in the posttest, it increased to 12.18 (standard deviation: 1.87). This increase in average scores indicates that the CAL program implemented during the intervention had a positive effect on students' performance. However, to ensure that this difference is not due to random factors but is statistically significant, further statistical analysis was conducted. The results of the analysis indicate a mean difference of

-4, with a standard deviation of 1.23. The t-value was calculated at -15.2 (df = 21), while the p-value was less than any common level of significance ($p < 0.001$). This confirms that the difference in the mean scores of the two variables is statistically significant, validating the positive impact of the CAL program. Based on these findings, it is evident that the educational intervention significantly contributed to improving students' performance.

The results indicate that the teaching intervention incorporating the CAL program had a statistically significant impact on enhancing associative thinking skills and fostering students' computational thinking abilities. The mean difference of -4.000 suggests that, on average, students' performance improved by 4 points from the pretest to the posttest. Furthermore, the extremely low p-value (0.000) demonstrates that this difference is highly unlikely to be due to random variation. Thus, we can conclude that the intervention significantly contributed to the development of students' computational thinking.

Table 4 Independent samples test

Pretest Score	Levene's Test for Equality of Variances		t-test for Equality of Means						
								95% CI	
	F	Sig.	t	df	Sig.(2-tailed)	MD	SE	Lower	Upper
Equal variances assumed	8.226	0.010	0.445	20	0.661	0.444	0.999	-1.640	2.529
Equal variances not assumed			0.402	11.492	0.695	0.444	1.105	-1.975	2.864

2.6.2 Analysis by age group

Next, we will examine the extent to which gender influenced students' performance in both the pretest and posttest. Initially, we will analyze whether gender played a role in students' performance during the pretest phase.

Table 5 Group statistics on posttest score

Gender	N	Mean	SD	SE
Boy	9	11.89	2.369	0.790
Girl	13	12.38	1.502	0.417

2.6.3 Analysis of variance equality and gender influence

The test for equality of variances yielded F = 8.226, with p = 0.01, which is less than the significance level of 0.05. Consequently, the null hypothesis, which assumes equal variances, is rejected. As a result, the t-test results for unequal variances ("Equal variances not assumed") are used. According to these results: t = 0.402 with degrees of freedom df = 11.49 and p = 0.695, which is greater than the significance level of 0.05.

Therefore, the null hypothesis (H_0) is not rejected, indicating no significant difference in the pretest mean scores between boys and girls. In conclusion, the t-test results show no statistically significant difference in pretest performance between boys and girls. The mean difference is minimal and not significant. A similar data analysis follows to determine whether and to what extent gender influenced children's performance in the posttest.

Table 6 Independent samples test

Posttest Score	Levene's Test for Equality of Variances		t-test for Equality of Means						
								95% CI	
	F	Sig.	t	df	Sig. (2-tailed)	MD	SD	Lower	Upper
Equal variances assumed	2.383	0.138	-0.603	20	0.554	-0.496	0.823	-2.212	1.220
Equal variances not assumed			-0.555	12.432	0.589	-0.496	0.893	-2.433	1.442

2.6.4 Posttest analysis of variance equality and gender influence

According to the table and the graph above, the test for equality of variances yielded F = 2.383, with p = 0.138, which is greater than the significance level of 0.05. Thus, the null hypothesis, which assumes equal variances is not rejected. Consequently, the t-test results for equal variances assumed are used. These results indicate: t = -0.603 with degrees of freedom df = 20 and p = 0.554, which is also greater than the significance level of 0.05.

Therefore, the null hypothesis (H_0) is not rejected, suggesting no significant difference in the posttest mean scores between boys and girls. In conclusion, the t-test results indicate that there is no statistically significant difference in posttest performance between boys and girls. The mean difference is small and not significant. The fact that the p-value is greater than 0.05 suggests that any observed differences in performance between the two genders can be attributed to random variation rather than a systematic effect of gender.

Table 7 Group statistics on pretest score

Gender	N	Mean	SD	SE
Boy	11	9.090	2.343	0.707
Girl	11	7.270	1.849	0.557

2.6.5 Analysis based on students' age group

The following analysis examines whether and to what extent students' age group (preschoolers/kindergarteners) influenced their performance in both the pretest and the posttest. Initially, we will assess whether the age group played a role in students' performance during the pretest phase.

Table 8 Independent samples test

Pretest Score	Levene's Test for Equality of Variances		t-test for Equality of Means						
								95% CI	
	F	Sig.	t	df	Sig. (2-tailed)	MD	SD	Lower	Upper
Equal variances assumed	0.616	0.442	2.020	20	0.057	1.818	0.900	-0.059	3.695
Equal variances not assumed			2.020	18.973	0.058	1.818	0.900	-0.066	3.702

2.6.6 Analysis of age group influence on pretest performance

Examining the results of the t-test for the effect of age group on pretest performance, as shown in the table above, the test for equality of variances yielded F = 0.616, with p = 0.442, which is greater than the significance level of 0.05. Therefore, the null hypothesis, assuming equal variances, is not rejected. Consequently, the t-test results for equal variances assumed are used. These results indicate: t = 2.020 with degrees of freedom df = 20 and p = 0.057, which is also greater than the significance level of 0.05. As a result, the null hypothesis (H_0) is not rejected. The mean pretest scores between the two age groups of students do not exhibit a particularly significant difference.

Table 9 Group statistics on posttest score

Gender	N	Mean	SD	SE
Boy	11	12.270	2.149	0.648
Girl	11	12.090	1.640	0.495

Now, let's examine the results of the corresponding data analysis test to determine whether, and to what extent, the age group influenced the children's performance in the post-test.

Table 10 Independent samples test

Posttest Score	Levene's Test for Equality of Variances		t-test for Equality of Means						
								95% CI	
	F	Sig.	t	df	Sig. (2-tailed)	MD	SD	Lower	Upper
Equal variances assumed	1.823	0.192	0.223	20	0.826	0.182	0.815	-1.519	1.882
Equal variances not assumed			0.223	18.700	0.826	0.182	0.815	-1.526	1.890

The results of the t-test for the effect of age group on pretest performance, as shown in the table above, indicate that the test for equality of variances yielded F = 1.823, with p = 0.192, which is greater than the significance level of 0.05. Therefore, the null hypothesis, stating that the variances are equal, is not rejected. Consequently, the results from the t-test assuming unequal variances were used, which showed: t = 0.223 with degrees of freedom df = 20 and p = 0.826, a value significantly larger than the 0.05 significance level. Hence, the null hypothesis (H_0) is not rejected. No statistically significant difference was found between the mean performance of preschoolers and kindergarteners during the posttest process (refer to the graph below).

3 Discussion and conclusions

The research question of this study aimed to examine whether the implementation of the "Coding as Another Language (CAL)" program, developed by the DevTech Research Group, contributes to the improvement of computational thinking in kindergarten students. The results of the teaching intervention, through the CAL program, positively answer the research question, showing significant development in the children's computational thinking. This was observed through a comparison of the mean scores in the pretest and posttest. Age group and gender did not affect the children's performance. After the program implementation, improvements were observed in both boys and girls, as well as in preschoolers and kindergarteners. Difficulties encountered by some children, particularly with the use of repeat and directional tiles, were mainly resolved through exploratory learning and peer interaction. Although the sample size was small, the students in the group where the program was implemented shared their positive learning experiences with students from other groups. This created curiosity among both the students and teachers of the other school groups, leading to thoughts of expanding the program to the entire student body in future academic years.

The students were particularly enthusiastic about the ability to record and include the corresponding tile in the code they were creating. It is crucial to note the evident excitement and interest of the children, both when the teaching was conducted using the interactive board and when portable devices were used in groups. They eagerly anticipated the next session, and after completing the program, it was observed that during free playtime in the learning centers, there was increased activity in the digital corner of the classroom, with most children attempting to create digital scenarios based on the CAL program. Perhaps most importantly, in my view, was the connection of the program to the natural environment, either by transferring the digital stories they created into real-life situations or by creating digital stories inspired by real events. Mathematical and programming concepts were easily and enjoyably grasped, and it became evident that the young students started to reflect on the vocabulary they needed to enhance their scenarios with sound.

Conflicts of interest

The author declares there is no conflict of interest.

References

Arithm, Y. A. (2021). Programma Spoudon gia tin Prosholiki Ekpaideusi. Efimeris tis Kyverniseos tiw Ellinikis Dhmokratias, FEK 5961/B/17-12-2021. (In Greek)

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. ACM Inroads, 2(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bell, T., Henderson, T., & Roberts, J. (2018). Computational thinking and CS Unplugged. https://csunplugged.org

Bers, M. U. (2017). Coding as a Playground. Routledge. https://doi.org/10.4324/9781315398945

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education, 72, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. Computers & Education, 138, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. European Commission, JRC Science

for Policy Report, 68.
https://doi.org/10.2791/792158

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Canada (pp. 1-25).

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. Review of Educational Research, 87(4), 834–860.
https://doi.org/10.3102/0034654317710096

Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. Computers in Human Behavior, 92, 361–372.
https://doi.org/10.1016/j.chb.2018.11.043

Council Recommendation of 22 May 2018 on key competences for lifelong learning. (2018). Official Journal of the European Union, 189, 1-13.

Critten, V., Hagon, H., & Messer, D. (2021). Can Pre-school Children Learn Programming and Coding Through Guided Play Activities? A Case Study in Computational Thinking. Early Childhood Education Journal, 50(6), 969–981.
https://doi.org/10.1007/s10643-021-01236-8

Dovigo, F. (2021). The role of teachers' attitude towards the use of the tablet in the first-grade elementary classroom. International Journal of Education and Development using Information and Communication Technology (IJEDICT), 17(3), 234-248.

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. Journal of Computer Assisted Learning, 32(6), 576–593. Portico.
https://doi.org/10.1111/jcal.12155

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: past, present, and future. Spotlight on Statistics, 1, 1-35.
https://www.bls.gov

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. Computers & Education, 63, 87–97.
https://doi.org/10.1016/j.compedu.2012.11.016

Fessakis, G., Komis, V., Dimitracopoulou, A., & Prantsoudi, S. (2019). Overview of the computer programming learning environments for primary education. Review of Science, Mathematics and ICT Education, 13(1), 7-33.
https://doi.org/10.26220/rev.3140

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr. Proceedings of the 12th International Conference on Interaction Design and Children, 1–10.
https://doi.org/10.1145/2485760.2485785

Gaki, O., & Tzimogiannis, A. (2021). Study of the contribution of Scratch Jr programming to the development of simple problem solving skills in kindergarten children. Conferences of the Hellenic Scientific Association of Information and Communication Technologies in Education, 458-465. ISBN: 978-618-83186-5-6. (in Greek)

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y.-A. (2019). Robotics to develop computational thinking in early Childhood Education. Comunicar, 27(59), 63–72.
https://doi.org/10.3916/c59-2019-06

Grover, S. (2018). The 5th 'C' of 21st century skills? Try computational thinking (not coding).
https://www.edsurge.com

Grover, S. (2018). The 5th 'C'of 21st century skills? Try computational thinking (not coding).

Gs/21072β. (2003). Diathematiko Eniaio Plaisio Programmaton Spoudon kai Analytika Programmata Prosholikis Agogis.

Heikkilä, M., & Mannila, L. (2018). Debugging in Programming as a Multimodal Practice in Early Childhood Education Settings. Multimodal Technologies and Interaction, 2(3), 42.
https://doi.org/10.3390/mti2030042

Heljakka, K., & Ihamäki, P. (2019). Ready, Steady, Move! Coding Toys, Preschoolers, and Mobile Playful Learning. Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration, 68–79.
https://doi.org/10.1007/978-3-030-21817-1_6

Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting Education in "Educational" Apps. Psychological Science in the Public Interest, 16(1), 3–34.
https://doi.org/10.1177/1529100615569721

Insituto Tecnologias kai Ipologiston "Diofantos". (2019). Epimorfotiko Yliko gia tin epimorfosi ton ekpaideutikon sta kendra stirixis epimorfosis. Patra: Insituto Tecnologias Ipologiston kai ekdoseon "Diofantos". (in Greek)

International Society for Technology in Education (ISTE). (2011). Computational thinking in K–12 education teacher resources (2nd ed.).
https://www.iste.org

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. Mobile Learning Applications in Early Childhood Education, 82–100.
https://doi.org/10.4018/978-1-7998-1486-3.ch005

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2012). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. Early Childhood Education Journal, 41(4), 245–255.
https://doi.org/10.1007/s10643-012-0554-5

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. Journal of Educational Multimedia and Hypermedia, 21(4), 371-391.

Koutsouvanou, E. (2006). Merikes Apopsis gia to Diathematiko Eniaio Plaisio Spoudon (DEEPS). Sygxrono Nipiagogio, 53, 96-106. (In Greek)

Lin, S.-Y., Chien, S.-Y., Hsiao, C.-L., Hsia, C.-H., & Chao, K.-M. (2020). Enhancing Computational Thinking Capability of Preschool Children by Game-based Smart Toys. Electronic Commerce Research and Applications, 44, 101011.
https://doi.org/10.1016/j.elerap.2020.101011

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

Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. International Journal of Child-Computer Interaction, 32, 100396.
https://doi.org/10.1016/j.ijcci.2021.100396

McManis, L. D., & Gunnewig, S. B. (2012). Finding the education in educational technology with early learners. Young Children, 67(3), 14–24.

Misirli, A., & Komis, V. (2014). Robotics and Programming Concepts in Early Childhood Education: A Conceptual Framework for Designing Educational Scenarios. Research on E-Learning and ICT in Education, 99–118.
https://doi.org/10.1007/978-1-4614-6501-0_8

Murcia, K. J., & Tang, K. S. (2019). Exploring the multimodality of young children's coding. Australian Educational Computing, 34(1).

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. The Asia-Pacific Education Researcher, 28(5), 387–397.
https://doi.org/10.1007/s40299-019-00438-4

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Systematic Reviews, 10(1).
https://doi.org/10.1186/s13643-021-01626-4

Papadakis, S. (2020). Robots and Robotics Kits for Early Childhood and First School Age. International Journal of Interactive Mobile Technologies (IJIM), 14(18), 34.
https://doi.org/10.3991/ijim.v14i18.16631

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. International Journal of Technology Enhanced Learning, 11(3), 231.
https://doi.org/10.1504/ijtel.2019.100478

Papadakis, S., & Kalogiannakis, M. (2020). A Research Synthesis of the Real Value of Self-Proclaimed Mobile Educational Applications for Young Children. Mobile Learning Applications in Early Childhood Education, 1–19.
https://doi.org/10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Alexandraki, F., & Zaranis, N. (2021). Mobile device use among preschool-aged children in Greece. Education and Information Technologies, 27(2), 2717–2750.
https://doi.org/10.1007/s10639-021-10718-6

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. International Journal of Mobile Learning and Organisation, 10(3), 187.
https://doi.org/10.1504/ijmlo.2016.077867

Papadakis, S., Vaiopoulou, J., Sifaki, E., Stamovlasis, D., Kalogiannakis, M., & Vassilakis, K. (2021). Factors That Hinder in-Service Teachers from Incorporating Educational Robotics into Their Daily or Future Teaching Practice. Proceedings of the 13th International Conference on Computer Supported Education.
https://doi.org/10.5220/0010413900550063

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

Petousi, V., & Sifaki, E. (2020). Contextualising harm in the framework of research misconduct. Findings from discourse analysis of scientific publications. International Journal of Sustainable Development, 23(3/4), 149.
https://doi.org/10.1504/ijsd.2020.115206

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education, 128, 52–62.
https://doi.org/10.1016/j.compedu.2018.09.006

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2015). Constructing the ScratchJr programming language in the early childhood classroom. International Journal of Technology and Design Education, 26(4), 489–504. https://doi.org/10.1007/s10798-015-9325-0

Pugnali, A., Sullivan, A., & Umashi Bers, M. (2017). The Impact of User Interface on Young Children's Computational Thinking. Journal of Information Technology Education: Innovations in Practice, 16, 171–193. https://doi.org/10.28945/3768

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. Journal of Science Education and Technology, 29(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x

Roussou, E., & Rangoussi, M. (2019). On the Use of Robotics for the Development of Computational Thinking in Kindergarten: Educational Intervention and Evaluation. Robotics in Education, 31–44. https://doi.org/10.1007/978-3-030-26945-6_3

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2019). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. The Asia-Pacific Education Researcher, 29(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Skaraki, E., Kalogiannakis, M., Ampartzaki, M., & Papadakis, S. (2018). Teaching natural science concepts to young children with mobile devices and hands-on activities. A case study. International Journal of Teaching and Case Studies, 9(2), 171. https://doi.org/10.1504/ijtcs.2018.10011893

Strawhacker, A., & Bers, M. U. (2018). What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children. Educational Technology Research and Development, 67(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Strawhacker, A., Lee, M., & Bers, M. U. (2017). Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. International Journal of Technology and Design Education, 28(2), 347–376. https://doi.org/10.1007/s10798-017-9400-9

Sullivan, A., & Bers, M. U. (2018). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. International Journal of Technology and Design Education, 29(5), 1033–1051. https://doi.org/10.1007/s10798-018-9483-y

Vaiopoulou, J., Papadakis, S., Sifaki, E., Stamovlasis, D., & Kalogiannakis, M. (2021). Parents' Perceptions of Educational Apps Use for Kindergarten Children: Development and Validation of a New Instrument (PEAU-p) and Exploration of Parents' Profiles. Behavioral Sciences, 11(6), 82. https://doi.org/10.3390/bs11060082

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2020). Teacher's Role in Fostering Preschoolers' Computational Thinking: An Exploratory Case Study. Early Education and Development, 32(1), 26–48. https://doi.org/10.1080/10409289.2020.1759012

Wing, J. (2011). Research notebook: Computational thinking—What and why. The link magazine, 6.

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

Xiao, Y., & Watson, M. (2017). Guidance on Conducting a Systematic Literature Review. Journal of Planning Education and Research, 39(1), 93–112. https://doi.org/10.1177/0739456x17723971

Yong Khoo, K. (2020). A Case Study on How Children Develop Computational Thinking Collaboratively with Robotics Toys. International Journal of Educational Technology and Learning, 9(1), 39–51. https://doi.org/10.20448/2003.91.39.51

Zaranis, N., & Alexandraki, F. (2019). Use of Tablets in Kindergarten for Teaching Multiplication by Using Models Based on Realistic Mathematics. In Education, Lifelong Learning, Research and Technological Development, Innovation and Economy, 2, 87. National Documentation Centre (ECT). https://doi.org/10.12681/elrie.1513

Zaranis, N., Kalogiannakis, M., & Papadakis, S. (2013). Using Mobile Devices for Teaching Realistic Mathematics in Kindergarten Education. Creative Education, 04(07), 1–10. https://doi.org/10.4236/ce.2013.47a1001